



UNIVERSITI MALAYA

Perpustakaan SKTM

**SISTEM PEMILIHAN CONTOH  
PENGATURCARAAN C (S2PC)**

Disediakan oleh:

**MOHD ZULKHAIRI BIN BAHARI**

**WEK 990169**

**SESI 2002/2003**

**PROJEK ILMIAH  
TAHAP AKHIR 2  
WXES 3182**

Penyelia:

**DR. RUKAINI HJ ABDULLAH**

Moderator:

**CIK NORISMA IDRIS**

---

## ABSTRAK

Pembelajaran berdasarkan contoh merupakan kaedah yang sering digunakan pelajar dalam menyelesaikan tugas yang diberikan oleh pensyarah. Namun dalam pembelajaran bahasa pengaturcaraan C pelajar-pelajar didapati sukar untuk mencari contoh rujukan yang sesuai untuk dijadikan panduan. Sistem Pemilihan Contoh Pengaturcaraan C (S2PC) dibangunkan untuk mengatasi masalah ini di mana contoh program yang sesuai akan dicadangkan kepada pelajar, berdasarkan ciri-ciri bahasa C yang dikehendaki mereka. Kaedah *case-based reasoning* digunakan dalam pembangunan sistem ini di mana kaedah ini menggunakan kes-kes lalu yang disimpan di dalam pangkalan pengetahuan untuk menyelesaikan masalah.

## PENGHARGAAN

Alhamdulillah, dengan izinNya dapat juga saya menyiapkan projek Latihan Ilmiah ini dalam masa yang telah ditetapkan. Di kesempatan ini saya ingin memberikan sekalung penghargaan kepada pihak-pihak yang telah banyak membantu saya dalam menyiapkan projek ini.

Pertama sekali, jutaan terima kasih kepada Dr Rukaini Hj Abdullah yang sudi menjadi penyelia kepada projek ini dan Cik Norisma Idris yang sudi untuk menjadi moderator. Terima kasih di atas segala nasihat dan bantuan yang diberikan.

Penghargaan tidak terhingga dilimpahkan kepada ibu bapa serta keluarga di atas berkat doa dan sokongan dari segi moral dan kewangan yang diberikan.

Terima kasih juga kepada Nik dan Fazli kerana di atas segala bantuan dan nasihat yang diberikan terhadap projek ini.

Akhir sekali, kepada sesiapa yang namanya tidak nyatakan di sini namun telah terlibat secara langsung atau tidak langsung di dalam projek ini, diucapkan jutaan terima kasih di atas bantuan anda.

## SENARAI RAJAH

NAMA RAJAH	HALAMAN
Rajah 1.1 : Peratusan keutamaan kaedah yang digunakan oleh pelajar untuk melakukan tugas	4
Rajah 1.2 : Peratusan keutamaan kaedah yang digunakan oleh pelajar FSKTM untuk membuat pengaturcaraan	4
Rajah 1.3 : Jadual Perancangan Pembangunan Projek	11
Rajah 2.1 : Kitar CBR	14
Rajah 2.2 : Struktur <i>task-methods</i> dalam CBR	17
Rajah 2.3 : Antaramuka sistem ELMPE	24
Rajah 3.1 : Peratusan agihan borang mengikut fakulti	28
Rajah 3.2 : Model air terjun dengan pemprototaipan	32
Rajah 4.1 : Rekabentuk sistem SPCPP = Kitar CBR	37
Rajah 4.2 : Ciri-ciri program (kes) di dalam pangkalan pengetahuan	38
Rajah 4.3 : Carta struktur Sistem Pemilihan Contoh Program Prolog	40
Rajah 4.7 : Antaramuka untuk menu utama S2PC	45
Rajah 4.8 : Antaramuka untuk modul pemilihan	45
Rajah 4.9 : Antaramuka untuk keputusan pemilihan	46



Rajah 6.1 : Punca-punca ralat semasa pembangunan	57
Rajah 6.2 : Langkah-langkah pengujian	60
Rajah 6.3 : Pengujian bawah atas	62

## SENARAI JADUAL

Abstrak

### NAMA JADUAL

### HALAMAN

Senarai Rajah

iii

Jadual 4.1: Perwakilan sebahagian kes dan ciri di dalam pangkalan

39

Kandungan data

vi

Jadual 5.1 : Jadual pepadanan nilai untuk dilakukan case-scoring

52

1.1 Pengiraan Projek

1

1.2 Pemastianan Yang Wujud

2

1.2.1 Penilaian Berdasarkan Ciri

2

1.2.2 Pemastianan Mempunyai Status C

3

1.3 Penyelesaian Masalah

8

1.4 Definisi Projek

6

1.5 Stop Projek

7

1.6 Objektif Projek

8

1.7 Matriks Projek

9

1.8 Penyelesaian Projek

10

## BAB 2 : KAJIAN LITERASI

2.1 Case Based Reasoning (CBR)

12

2.1.1 Apakah case based reasoning?

12

2.1.2 Kiri-CBR

12

## ISI KANDUNGAN

Abstrak	i
Penghargaan	ii
Senarai Rajah	iii
Senarai Jadual	v
Kandungan	vi

### BAB 1 : PENGENALAN

1.1	Pengenalan Projek	1
1.2	Permasalahan Yang Wujud	2
1.2.1	Pembelajaran Berdasarkan Contoh	2
1.2.2	Permasalahan Mempelajari Bahasa C	5
1.3	Penyelesaian Masalah	6
1.4	Definisi Projek	6
1.5	Skop Projek	7
1.6	Objektif projek	8
1.7	Matlamat Projek	9
1.8	Perancangan Projek	10

### BAB 2 : KAJIAN LITERASI

2.1	Case-Based Reasoning (CBR)	12
2.1.1	Apakah case-based reasoning?	12
2.1.2	Kitar CBR	12

2.1.3	CBR task-methods	16
2.1.4	Isu-isu Dalam Pembangunan Sistem CBR	18
2.1.5	Kelebihan Sistem CBR	20
2.1.6	Kelemahan Sistem CBR	20
2.2	Algoritma Nearest Neighbour	21
2.3	Kajian Sstem Sedia Ada	22
2.3.1	ELMPE	22
2.3.2	Kelebihan ELMPE	23
2.3.3	Kekurangan ELMPE	24
2.4	Sumbangan dan Perbezaan S2PC	25

### **BAB 3 : METODOLOGI PEMBANGUNAN SISTEM**

3.1	Kaedah Mendapatkan Penyelesaian	27
3.2	Metodologi Pembangunan Sistem	29
3.2.1	Metodologi Air Terjun dengan Prototaip	30
3.2.2	Kelebihan Model Air Terjun dengan Prototaip	33
3.3	Keperluan Perisian	33
3.4	Keperluan Perkakasan	34

### **BAB 4 : REKABENTUK SISTEM**

4.1	Rekabentuk SPCPP	36
4.2	Rekabentuk Pangkalan Data	37



4.3	Rekabentuk Struktur	40
4.3.1	Modul yang terlibat	40
4.4.2	Keperluan Kefungsian	41
4.4.3	Keperluan Bukan Fungsian	42
4.4	Rekabentuk Antaramuka	43
4.5	Penilaian Rekabentuk Sistem	46
<b>BAB 5 : PENGKODAN DAN IMPLEMENTASI</b>		49
5.1	Pengkodan dan Implementasi S2PC	49
5.1.1	Kandungan dan Antaramuka S2PC	49
5.1.2	Pengintegrasian Pangkalan Pengetahuan	51
5.1.3	Pengkodan Case-scoring	51
5.2	Rumusan	55
<b>BAB 6 : PENGUJIAN PROGRAM DAN SISTEM</b>		56
6.1	Tujuan Pengujian	56
6.1.1	Jenis-jenis Ralat dan Kegagalan	56
6.2	Langkah-langkah Pengujian	59
6.2.1	Ujian Unit	60
6.2.2	Ujian Integrasi	61
6.2.3	Ujian Fungsi	62
6.2.4	Ujian Prestasi	63

6.2.5	Ujian Penerimaan	63
6.2.6	Ujian Pemasangan	63
6.3	Rumusan	64

<b>BAB 7 : PENILAIAN SISTEM</b>		65
7.1	Pencapaian Objektif	65
7.2	Masalah Pembangunan Sistem dan Penyelesaian	67
7.3	Penilaian Pengguna	69
7.4	Kekangan Sistem	70
7.5	Cadangan Masa Hadapan	72
7.6	Rumusan	74

<b>RUJUKAN</b>	75
----------------	----

<b>LAMPIRAN</b>	78
-----------------	----



## **BAB 1                    PENGENALAN**

### **1.1    PENGENALAN PROJEK**

Penggunaan komputer dalam sistem pendidikan pada masa kini adalah sangat meluas iaitu daripada peringkat pengurusan dan pentadbiran hinggalah kepada peringkat pengajaran dan pembelajaran. Pelbagai sistem telah dibina serta dibangunkan dalam mempertingkatkan kecekapan serta menaikkan mutu pendidikan. Pembelajaran menggunakan komputer adalah antara sistem yang telah dibina di mana ia menjadi alat bantuan kepada pelajar mempelajari sesuatu kursus atau matapelajaran.

Sistem Pemilihan Contoh Pengaturcaraan C (S2PC) juga berkaitan dengan pembelajaran dan pendidikan. Sistem yang akan dibangunkan ini adalah berkaitan dengan pembelajaran program pengaturcaraan konvensional iaitu C, sebuah program pengaturcaraan yang digunakan dengan meluas dan sering dijadikan asas untuk pelajar-pelajar dalam bidang Sains Komputer.

Secara amnya S2PC memberi penekanan kepada pembelajaran kursus WXES 1108 Asas Pengaturcaraan Komputer di Fakulti Sains Komputer dan Teknologi Maklumat (FSKTM), Universiti Malaya. Pengguna S2PC ini disasarkan kepada pelajar-pelajar tahun satu serta mereka yang ingin memilih contoh-contoh program C yang sesuai sebagai bahan rujukan untuk menyelesaikan masalah

atau tugas yang diberi. Maka sistem ini secara tepatnya boleh dikatakan sebagai alat bantu pembelajaran yang sesuai digunakan oleh pelajar-pelajar untuk meningkatkan pemahaman mengenai bahasa C.

Projek ini menggunakan satu kaedah yang dikenali sebagai *case-based reasoning (CBR)* iaitu salah satu kaedah kepintaran buatan dalam pencarian data di mana pangkalan pengetahuan sistem akan terdiri daripada kes-kes yang lalu. Pangkalan pengetahuan ini akan terdiri daripada contoh-contoh program C yang betul dan boleh dilarikan. Sistem CBR akan membuat perbandingan program-program yang terdapat di dalam pangkalan pengetahuan dan seterusnya memilih contoh program yang paling sesuai mengikut ciri yang dikehendaki pengguna.

## **1.2 PERMASALAHAN YANG WUJUD**

### **1.2.1 Pembelajaran berdasarkan contoh**

Pada masa kini terdapat pelbagai kajian, seminar, kertas kerja dan sebagainya telah dilakukan bagi mengenalpasti kaedah yang sesuai dalam mempertingkatkan lagi mutu pembelajaran di institusi pengajian tinggi. Kajian ini adalah meluas daripada bagaimana kaedah terbaik untuk mengajar hinggalah kaedah yang sesuai digunakan untuk pembelajaran.



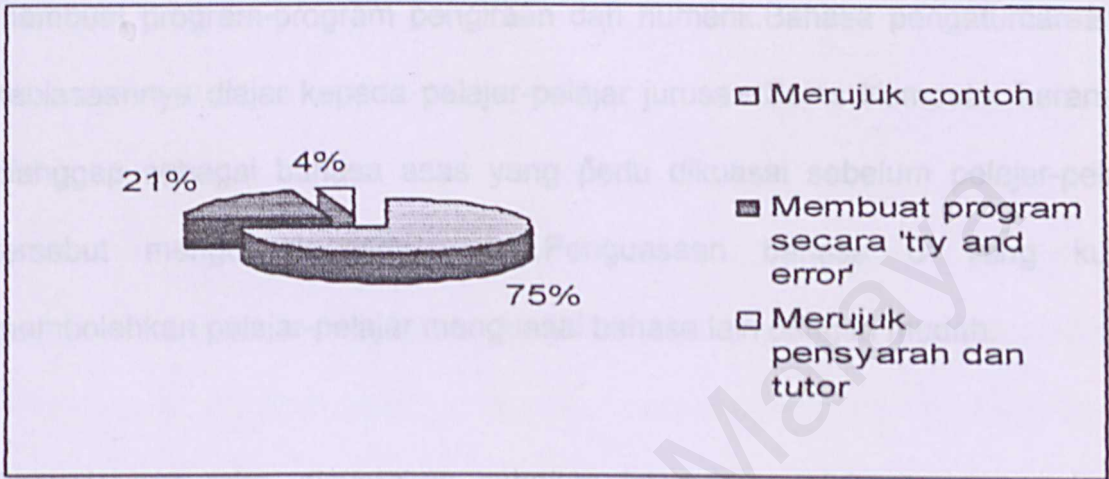
Biasanya apabila pelajar-pelajar diberikan tugas, mereka cenderung untuk merujuk contoh-contoh penyelesaian yang lalu sebagai rujukan bagi menyelesaikan tugas tersebut. Kajian Caellar [7] membuktikan perkara apabila dia menyatakan bahawa pelajar perlu diberikan contoh-contoh sesuai sebagai rujukan dalam meningkatkan lagi pemahaman mereka .

Satu soal-selidik juga telah dijalankan bagi mengkaji serta membuktikan sejauh manakah penggunaan kaedah merujuk contoh dalam menyelesaikan tugas, digunakan oleh para pelajar. Pensampelan bagi soal-selidik dijalankan mengikut kaedah pensampelan mudah, di mana borang soal-selidik diedarkan kepada responden-responden yang mudah diperolehi daripada pelbagai fakulti dan majoriti borang diedarkan kepada pelajar FSKTM.

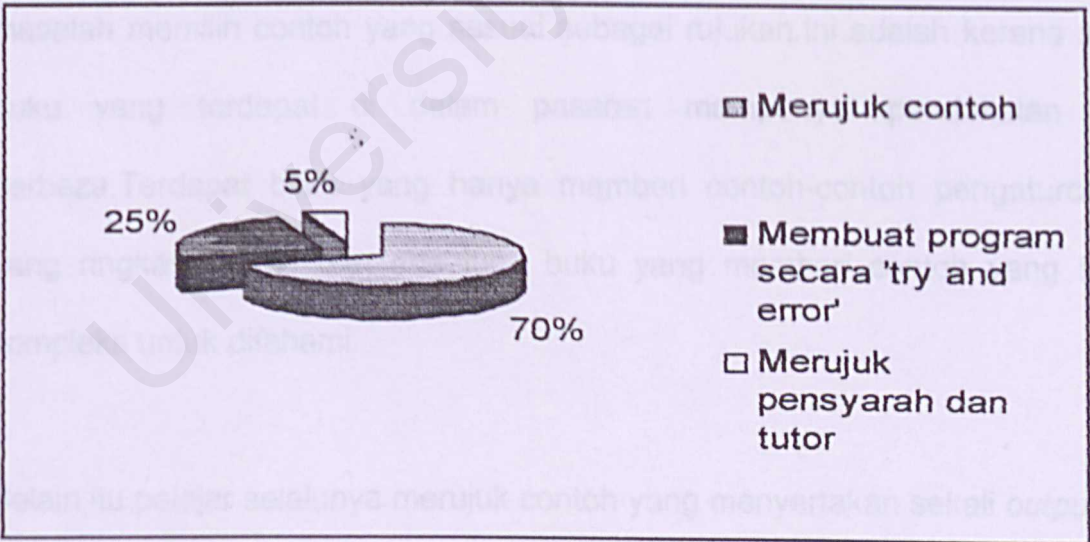
Berdasarkan analisis soalan pertama telah didapati bahawa kesemua pelajar dari pelbagai fakulti mengakui menggunakan kaedah penggunaan contoh-contoh yang sedia ada di dalam apa-apa jenis bentuk rujukan. Analisis soalan 3 pula membuktikan merujuk contoh ,menjadi keutamaan dalam melakukan tugas yang diberi. Rajah 1.1 menunjukkan peratusan keutamaan kaedah yang digunakan oleh pelajar untuk melakukan tugas.

Pembuktian bahawa merujuk contoh menjadi kaedah utama pelajar-pelajar FSKTM menyelesaikan tugas berkaitan pengaturcaraan, boleh dilihat

daripada analisis soalan 8 soal-selidik. Rajah 1.2 menunjukkan peratusan keutamaan kaedah yang diguna pelajar FSKTM untuk membuat pengaturcaraan.



Rajah 1.1: Peratusan keutamaan kaedah yang digunakan oleh pelajar untuk melakukan tugas.



Rajah 1.2: Peratusan keutamaan kaedah yang diguna pelajar FSKTM untuk membuat pengaturcaraan



### 1.2.2 Permasalahan Mempelajari Bahasa C

Bahasa merupakan bahasa konvensional yang biasanya digunakan untuk membuat program-program pengiraan dan numerik. Bahasa pengaturcaraan C kebiasaannya diajar kepada pelajar-pelajar jurusan Sains Komputer kerana ia dianggap sebagai bahasa asas yang perlu dikuasai sebelum pelajar-pelajar tersebut menguasai bahasa lain. Penguasaan bahasa C yang kukuh membolehkan pelajar-pelajar menguasai bahasa lain dengan mudah.

Seperti yang telah dinyatakan sebelum ini pelajar-pelajar cenderung untuk merujuk contoh apabila diberi tugas. Walaupun terdapat banyak buku rujukan berkaitan dengan pengaturcaraan C di pasaran, pelajar turut mengalami masalah memilih contoh yang sesuai sebagai rujukan. Ini adalah kerana setiap buku yang terdapat di dalam pasaran mempunyai pendekatan yang berbeza. Terdapat buku yang hanya memberi contoh-contoh pengaturcaraan yang ringkas sahaja dan ada juga buku yang memberi contoh yang terlalu kompleks untuk difahami.

Selain itu pelajar selalunya merujuk contoh yang menyertakan sekali *output* bagi program yang dinyatakan. Namun, buku-buku yang berada di pasaran ada yang memberikan contoh *output* dan ada yang tidak. Maka seseorang pelajar yang

ingin menguasai pengaturcaraan C biasanya terpaksa menggunakan lebih daripada satu atau dua buku untuk mendapatkan rujukan yang secukupnya.

### 1.3 PENYELESAIAN MASALAH

Berdasarkan permasalahan yang dinyatakan iaitu pembelajaran berdasarkan contoh dan permasalahan dalam C, maka dicadangkan satu penyelesaian iaitu pembinaan satu sistem yang menggunakan konsep pembelajaran berdasarkan contoh untuk mengatasi masalah dalam pembelajaran C. Sistem Pemilihan Contoh Pengaturcaraan C (S2PC) adalah sistem dicadangkan di mana ia akan menggunakan kaedah *case-based reasoning* sebagai kaedah penyelesaian masalah. Sistem ini akan mencadangkan contoh yang sesuai untuk dijadikan panduan berdasarkan ciri-ciri di dalam bahasa C yang dinyatakan pelajar dan contoh tersebut dinyatakan sekali *output* yang akan dihasilkan bagi program tersebut apabila dilarikan.

### 1.4 DEFINISI PROJEK

Perkataan sistem dalam bidang perkomputeran adalah ditakrifkan sebagai satu koleksi komponen-komponen berkaitan yang saling berinteraksi untuk menerima input, memproses, serta menghasilkan output yang digunakan bagi mencapai



sesuatu objektif [16]. Bahasa C pula merupakan sejenis bahasa pengaturcaraan paras tinggi yang digunakan untuk menulis perisian sistem atau aplikasi sistem[14].

Berdasarkan takrifan dua perkara di atas maka dapat dijelaskan bahawa S2PC adalah suatu sistem yang akan menerima input daripada pengguna dan menghasilkan output yang merupakan contoh program bahasa pengaturcaraan C. Secara dasarnya pengguna akan memasukkan input yang merupakan ciri-ciri yang dikehendaki dalam program C yang ingin dijadikan rujukan oleh pelajar dalam melakukan tugas. Melalui kaedah di dalam CBR sistem ini seterusnya akan memilih contoh program di dalam pangkalan pengetahuan berdasarkan ciri-ciri yang dinyatakan, dan kemudiannya memaparkan contoh program yang paling sesuai untuk dirujuk.

## 1.5 SKOP PROJEK

Sistem yang dicadangkan ini mempunyai skop-skop pembangunan yang tertentu iaitu :

### ➤ Ciri-ciri sistem

Sistem yang akan dibangunkan ini adalah satu sistem pintar yang menggunakan kaedah *case-based reasoning*.

➤ **Pembelajaran C**

Melalui sistem ini program-program pengaturcaraan yang akan dipilih oleh sistem dalam membantu pelajar-pelajar adalah terhad kepada topik-topik asas dalam pembelajaran C mengikut kursus WXES 1108 Asas Pengaturcaraan C.

➤ **Bahasa**

Bahasa yang akan digunakan di dalam sistem ini ialah Bahasa Melayu, memandangkan bahasa ini lebih mudah difahami pelajar.

## 1.6 OBJEKTIF PROJEK

Pembangunan projek ini mempunyai beberapa objektif yang disasarkan iaitu:

➤ **Sistem pintar case-based reasoning**

Objektif utama projek ini ialah penggunaan *case-based reasoning* sebagai kaedah yang berkesan dalam menyelesaikan permasalahan-permasalahan yang telah dinyatakan.

➤ **Memudahkan proses pemilihan contoh program**

Melalui sistem ini, pelajar-pelajar akan memperolehi kemudahan untuk memilih contoh program yang sesuai untuk dirujuk. Sistem ini akan



- menjadi jawapan kepada masalah pelajar yang perlu mencari pelbagai jenis buku rujukan untuk mencari contoh program C yang sesuai.

#### ➤ **Meningkatkan pencapaian pelajar**

Penggunaan sistem ini juga disasarkan sebagai salah satu cara dalam meningkatkan pencapaian pelajar dalam pembelajaran pengaturcaraan C.

### 1.7 MATLAMAT PROJEK

Pembangunan S2PC ini mensasarkan beberapa matlamat yang ingin dicapai setelah sistem ini siap serta diguna pakai iaitu:

#### ➤ **Penggunaan CBR lebih meluas**

Pembangunan sistem ini yang menggunakan kaedah case-based reasoning diharapkan akan menggalakkan lagi penggunaan CBR sebagai enjin penyelesaian masalah.

#### ➤ **C lebih difahami pelajar**

Pengguna sistem ini dijangkakan akan lebih memahami bahasa C memandangkan kemudahan untuk mereka mendapatkan rujukan contoh pengaturcaraannya.

➤ **Peningkatan pencapaian dalam kursus WXES 1108**

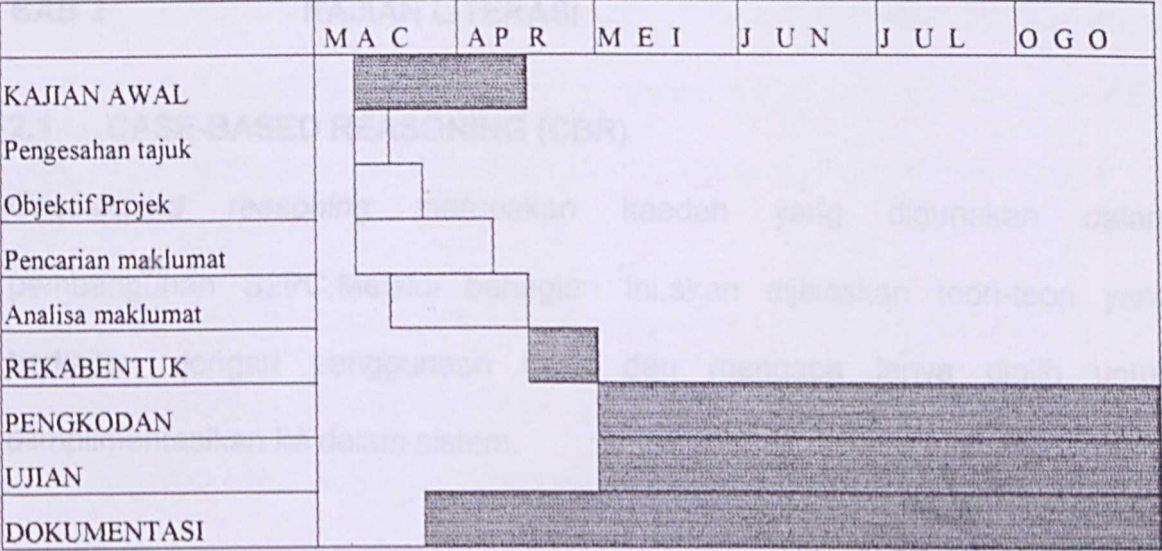
Seiring dengan kemudahan yang ditawarkan oleh sistem ini, maka diharapkan pencapaian pelajar-pelajar dalam kursus WXES 1108 khususnya serta pembelajaran C amnya, akan meningkat.

## **1.8 PERANCANGAN PROJEK**

Projek ini dimulakan pada semester khas sesi 2001/2002. Tarikh pengesahan tajuk ialah pada 15 Mac 2001. Projek ini dibahagikan kepada dua fasa dimana fasa satu akan dilaksanakan pada semester khas sesi 2001/2002 dan fasa dua dilaksanakan pada semester satu sesi 2002/2003. Fasa satu melibatkan Analisa Sistem dan Rekabentuk Sistem. Analisa Sistem dimulakan pada 19 Mac 2002 dan akan berakhir pada 14 April 2002. Rekabentuk Sistem pula dijalankan pada 15 April 2001 dan berakhir pada 3 Mei 2002.

Fasa dua melibatkan proses pengkodan dan pengujian. Proses pengkodan telah bermula 4 Mei 2002 dan berakhir pada 23 Ogos 2002. Pengujian dan perlaksanaan sistem juga telah dimulakan pada 4 Mei 2002 iaitu serentak dengan proses pengkodan dan berakhir juga pada 23 Ogos 2002. Proses dokumentasi pula dilakukan sepanjang perjalanan projek. Rajah 1.1- menunjukkan Carta Gantt bagi keseluruhan projek.





Rajah 1.3 : Jadual Perancangan Pembangunan Projek ( Carta Gantt ).

## BAB 2 KAJIAN LITERASI

### 2.1 CASE-BASED REASONING (CBR)

*Case-based reasoning* merupakan kaedah yang digunakan dalam pembangunan S2PC. Melalui bahagian ini, akan dijelaskan teori-teori yang berkaitan dengan penggunaan CBR dan mengapa ianya dipilih untuk diimplimentasikan ke dalam sistem.

#### 2.1.1 Apakah Case-based Reasoning (CBR) ?

*Case-based reasoning (CBR)* merupakan salah pendekatan dalam penyelesaian masalah secara *knowledge problem solving* di dalam bidang Kepintaran Buatan [ 13 ]. Terdapat beberapa takrifan yang telah diberikan terhadap CBR antaranya ialah:

- CBR ialah suatu pendekatan penyelesaian masalah yang bergantung kepada kes lalu atau kes yang hampir serupa untuk mencari penyelesaian masalah, ubahsuai serta kritik penyelesaian sedia ada dan jelaskan situasi yang mempunyai perbezaan

[ 12 ].

- CBR adalah berdasarkan kepada idea menggunakan penyelesaian masalah lalu untuk selesaikan masalah baru [ 10 ]



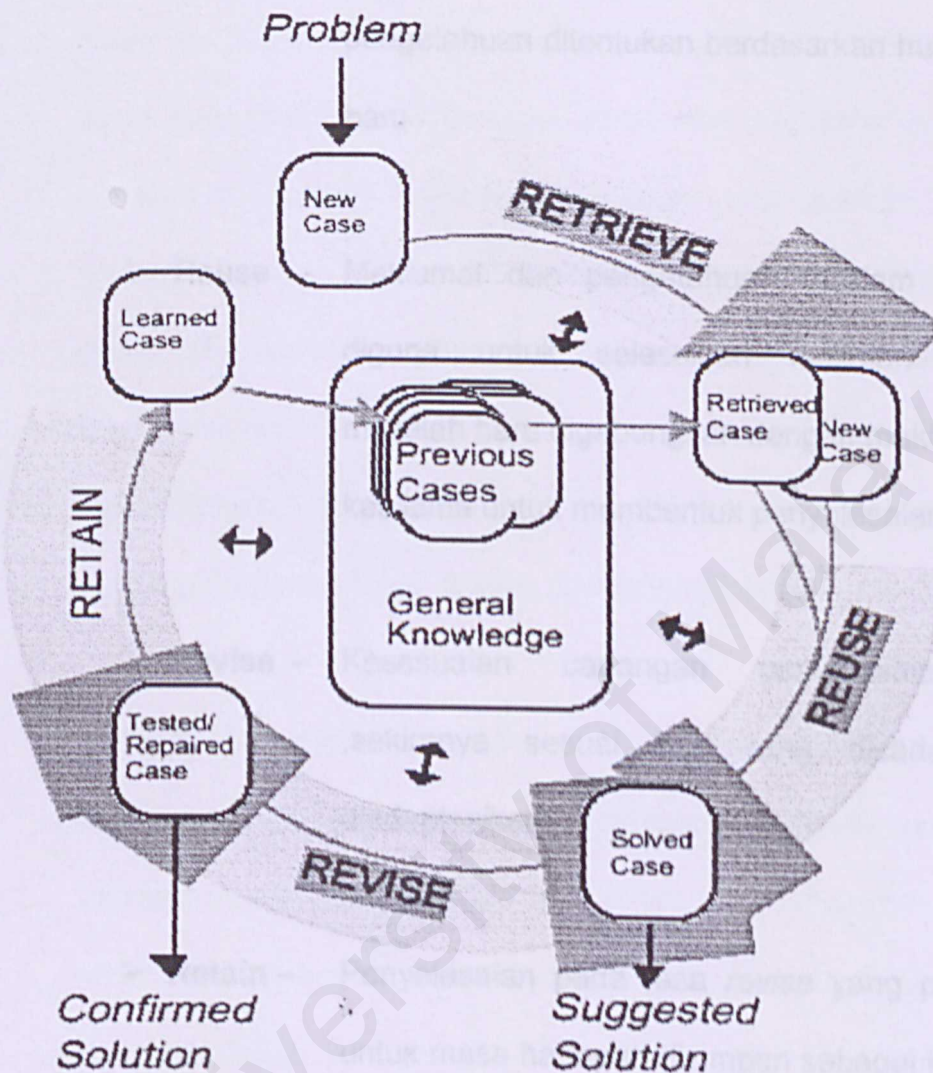
- CBR adalah satu kitaran paradigma penyelesaian masalah dalam Kepintaran Buatan, yang menekankan penggunaan semula penyelesaian sedia ada terhadap masalah yang hampir serupa [ 2 ].

Berdasarkan takrifan-takrifan di atas jelas bahawa CBR ini adalah berdasarkan konsep di mana manusia biasanya akan menggunakan pengalaman-pengalamannya yang lalu ketika berhadapan dengan situasi-situasi yang baru. Sebagai contoh seorang mekanik akan menggunakan pengalaman-pengalaman membaiki kereta yang lalu sebagai panduan dalam menentukan apakah masalah bagi kereta yang hendak diperbaikinya.

Konsep ini apabila dijadikan satu model kognitif yang menjelaskan bagaimana manusia mengguna dan membuat keputusan berdasarkan pengalaman kemudiannya digabungkan dengan teknologi untuk mempersembah dan mencari pengalaman tersebut maka wujudlah apa yang dikatakan sebagai CBR [ 12 ].

Pengaplikasian CBR di dalam pembangunan sistem pada kini adalah meliputi pelbagai bidang seperti perubatan, kejuruteraan, pendidikan dan termasuk sains komputer sendiri. Teknologi CBR kini semakin giat dikaji dan diimplimentasikan terutamanya di negara-negara maju.

### 2.1.2 Kitar CBR



Rajah 2.1: Kitar CBR [ Aamodt & Plaza, 1994]

Melalui rajah di atas Aamodt dan Plaza telah menjelaskan mengenai CBR dan proses-proses yang terdapat di dalamnya, di mana mereka telah menggambarkan CBR adalah satu proses kitaran yang mengandungi empat fasa iaitu :



- **Retrieve** - Kes yang hampir sama atau kes dalam pangkalan pengetahuan ditentukan berdasarkan huraian masalah baru
- **Reuse** - Maklumat dan pengetahuan didalam retrieve case diguna untuk selesaikan masalah baru. Huraian masalah baru digabungkan dengan maklumat di dalam kes lama untuk membentuk penyelesaian kes
- **Revise** - Kesesuaian cadangan penyelesaian dinilai, sekiranya sesuai kes yang dicadangkan perlu diadaptasikan.
- **Retain** - Penyelesaian pada fasa *revise* yang perlu disimpan untuk masa hadapan disimpan sebagai kes baru, *case base* dikemaskinikan dengan kes yang baru dipelajari.

*Previous cases* di dalam kitar CBR ini adalah merujuk kepada himpunan kes- kes lalu yang disimpan di dalam pangkalan pengetahuan sistem. *General knowledge* di dalam kitar ini menyokong proses CBR di mana ia mengandungi pengetahuan am berkaitan dengan domain sistem.

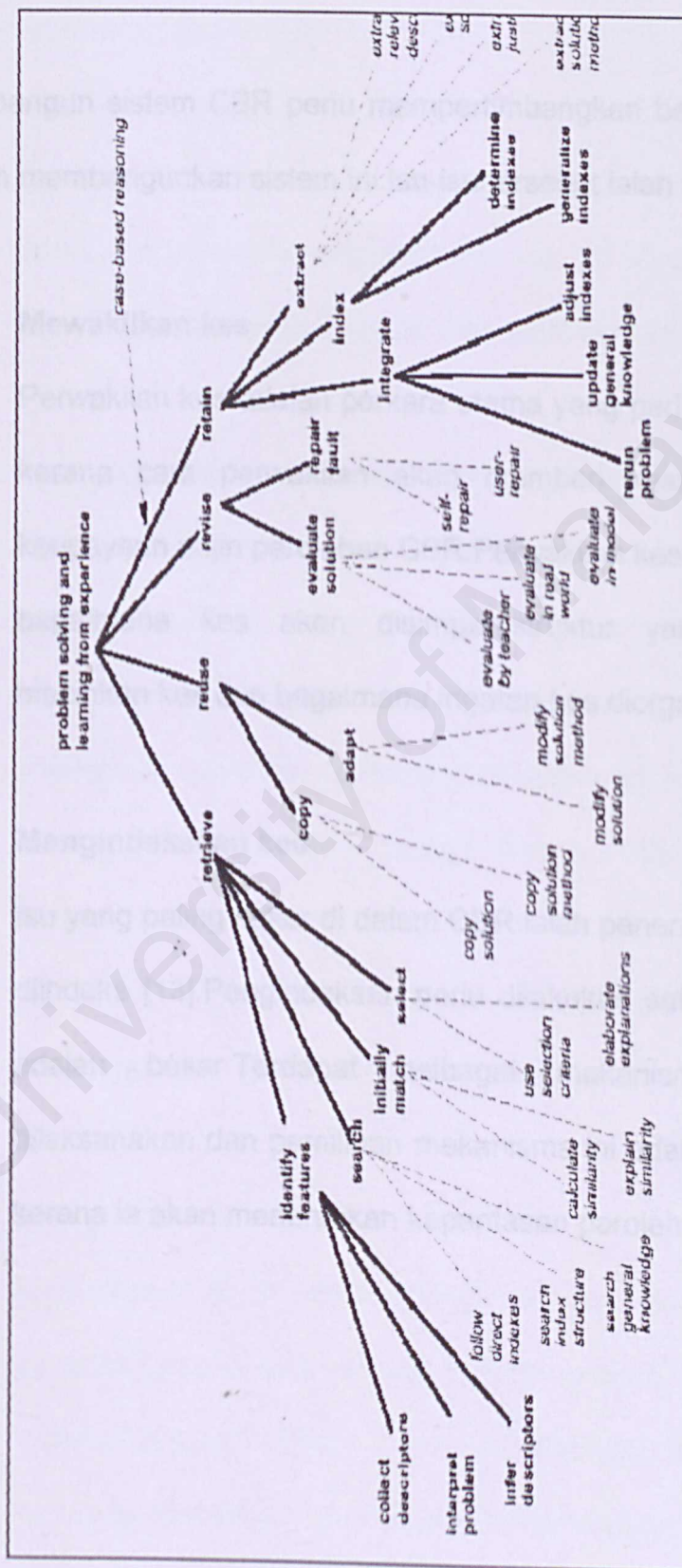
### 2.1.3 CBR *task-methods* [1]

Pada paras pengetahuan (*knowledge level*) sistem dilihat sebagai suatu agen yang mempunyai sasaran (*goal*) dan berusaha untuk mencapai sasaran itu. Sebuah sistem CBR boleh digambarkan daripada tiga perspektif iaitu tugas (*tasks*), kaedah (*methods*) dan model pengetahuan domain. Tugas adalah ditentukan oleh sasaran bagi sistem, dan tugas dilaksanakan dengan menggunakan beberapa kaedah. Bagi membolehkan kaedah-kaedah menjayakan tugas, ia memerlukan pengetahuan mengenai domain dan masalah terkini.

Struktur *task-methods* dalam suatu sistem CBR dapat dijelaskan berdasarkan rajah 2.2. Tugas dinyatakan dalam perkataan yang digelapkan manakala kaedah pula dinyatakan dalam tulisan italik. Rajah tersebut menjelaskan secara terperinci mengenai tugas dan kaedah yang terdapat dalam empat fasa kitar CBR iaitu *retrieve*, *revise*, *reuse* dan *retain*.

Garis-garis lurus di dalam rajah yang menghubungkan nod-nod tugas tersebut menggambarkan hubungan tugas dengan pecahan subtugasan-subtugasannya apabila menuruni pepohon. Manakala garis-garis terputus yang menghubungkan antara nod tugas dan nod kaedah, menunjukkan alternatif atau pilihan yang ada untuk menyiapkan tugas.





Rajah 2.2: Struktur task-methods dalam CBR [Aamodt & Plaza, 1994]



#### 2.1.4 Isu-isu dalam pembangunan sistem CBR

Pembangun sistem CBR perlu mempertimbangkan beberapa isu utama dalam membangunkan sistem ini. Isu-isu tersebut ialah :

➤ **Mewakilkkan kes**

Perwakilan kes adalah perkara utama yang perlu dipertimbangkan kerana cara perwakilan akan memberi kesan kepada tahap keupayaan enjin perolehan CBR. Perwakilan kes akan menentukan bagaimana kes akan disimpan, struktur yang sesuai untuk memecir kes dan bagaimana ingatan kes diorganisasikan.

➤ **Mengindekskan kes**

Isu yang paling sukar di dalam CBR ialah penentuan ciri kes untuk diindeks [13]. Pengindeksan perlu dilakukan sekiranya *case base* adalah besar. Terdapat pelbagai mekanisma yang boleh dilaksanakan dan pemilihan mekanisma ini adalah sangat penting kerana ia akan menentukan kepantasan perolehan kes.

- **Menaksirkan kesamaan antara kes terkini dengan kes yang diperoleh (*retrieved*).**

- Antara ciri pengenalan sesuatu sistem CBR ialah mekanisme menaksirkan kesamaan kes baru dengan kes di dalam *case base*. Kaedah menaksirkan persamaan ini adalah penting supaya sistem dapat memenuhi kehendak sebenar pengguna.

- **Mengadaptasikan penyelesaian lalu dengan masalah baru**

- Dalam fasa *retrieve*, apabila hanya beberapa bahagian sahaja huraian kes baru yang menyamai kes lama, sistem akan cuba melakukan adaptasi terhadap kes yang diperolehi untuk memenuhi kehendak semasa [18]. Sesetengah sistem CBR tidak menggunakan fungsi ini sebagai contoh sistem yang dibina oleh McIvor & Humphreys [15] memandangkan prosesnya yang kompleks. S2PC juga tidak menggunakan fungsi adaptasi ini memandangkan tempoh pembangunan yang singkat.

- **Mengorganisasikan CBR**

- Organisasi CBR merujuk kepada cara kes diorganisasikan untuk capaian semasa fasa *retrieve* [18]. Pengorganisasian adalah perlu supaya sistem adalah konsisten. Sebagai contoh kes yang berulang dalam fasa *retain* tidak disimpan dalam *case base*.



### 2.1.5 Kelebihan sistem CBR

Sistem CBR mempunyai beberapa kelebihan iaitu :

- Mempercepatkan masa dalam mendapatkan penyelesaian
- Lebih mudah untuk menyediakan *case-based* berbanding dengan *rules*
- CBR menyediakan kemudahan untuk membina, menilai dan menyelenggara sistem
- Membenarkan sistem cadangkan penyelesaian untuk domain yang tidak difahami sepenuhnya atau kurang pengetahuan.
- Dapat memelihara kepakaran dan pengalaman di dalam sesuatu bidang
- Perolehan pengetahuan yang secara relatifnya lebih mudah berbanding kaedah penyelesaian masalah yang lain.

### 2.1.6 Kelemahan sistem CBR

Sistem ini juga mempunyai beberapa kelemahan iaitu :

- Pemilihan kaedah perolehan maklumat yang tidak sesuai akan melambatkan masa penyelesaian
- Sukar memilih kaedah pengindeksan dan pepadanan yang sesuai.
- Penambahan kes baru di dalam pangkalan pengetahuan mungkin akan menyebabkan berlakunya pengulangan kes



- Kekurangan kes-kes lalu menyukarkan sesuatu sistem dibangunkan.
- Penyelesaian kes yang lalu mungkin tidak sesuai lagi mengikut arus zaman.

## 2.2 ALGORITMA NEAREST NEIGHBOUR

Algoritma *nearest neighbour* merupakan kaedah yang digunakan bagi memperoleh kes di dalam S2PC. Kebanyakan sistem CBR yang dikaji menggunakan algoritma ini iaitu sistem yang dibina oleh Ribeiro [18], McIvor & Humphreys [15] dan Watson & Gardingen [20].

Algoritma ini berfungsi dengan mencapai *case base* melalui perbandingan nilai pemberat pada ciri-ciri dalam kes baru dengan nilai pemberat pada ciri-ciri di dalam kes dalam pangkalan pengetahuan. Berdasarkan nilai pemberat ciri-ciri tersebut, markah agregat bagi pepadanan dikira [ 12 ] :

$$S(P, C) = \frac{\sum_{i=1}^n W_i \times \text{sim}(P_i C_i)}{\sum_{i=1}^n W_i},$$

P = kes baru                      C = kes di dalam *case base*

S = kesamaan antara P dan C

$W_i$  = pemberat untuk ciri  $i$

$sim(P_i C_i)$  = kesamaan antara nilai ciri-ciri P dan C

Perolehan kes (*retrieved case*) yang mempunyai markah agregat yang paling tinggi akan mewakili pemadanan yang paling hampir, manakala kes yang mempunyai markah lebih rendah akan diletakkan di bawah kes yang lebih tinggi markahnya.

## 2.3 KAJIAN SISTEM SEDIA ADA

Hasil daripada penyelidikan tidak ditemui sistem yang meyamai sistem S2PC iaitu menggunakan CBR untuk membantu pelajar dalam pembelajaran C. Namun terdapat sebuah sistem yang menghampiri sistem S2PC iaitu sistem yang menggunakan *case-based* dalam membantu pelajar mempelajari LISP.

### 2.3.1 ELMPE

ELMPE adalah bermaksud Episodic Learner Model Programming Environment. Sistem ini telah dibangunkan oleh Jabatan Psikologi, University of Trier, Germany yang diketuai oleh Professor Dr. Gerhard Weber [21].



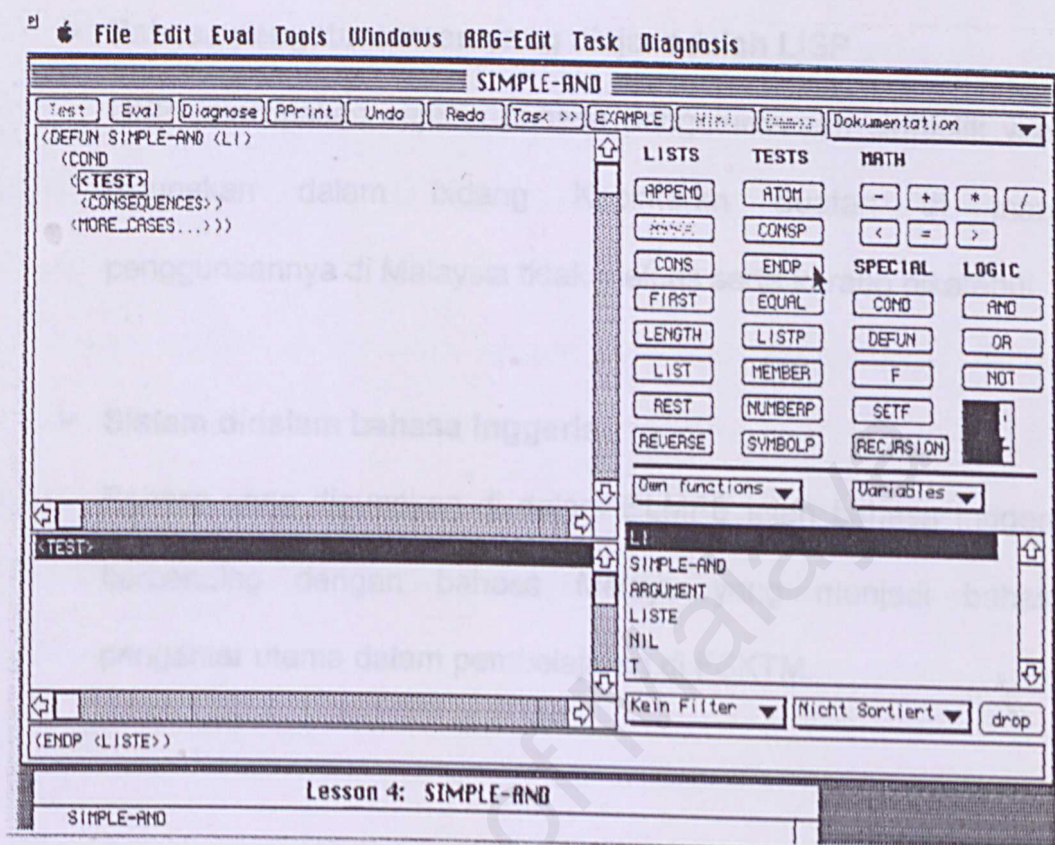
Sistem ini merupakan sebuah sistem pintar yang menggunakan kaedah *case-based* dalam pembangunan sistemnya. Sistem ini dibina untuk membantu mereka yang ingin mempelajari bahasa pengaturcaraan LISP, di mana LISP ini merupakan bahasa yang digunakan dalam bidang Kepintaran Buatan.

### 2.3.2 Kelebihan ELMPE

ELMPE adalah satu sistem pengaturcaraan terbuka di mana pelajar boleh membuat program pengaturcaraan LISP di dalamnya. Selain itu sistem ini juga mempunyai modul pembelajaran di mana pelajar dibenarkan melakukan latihan di dalam sistem dan juga boleh memilih latihan mengikut kehendak mereka. Disamping pelajar dibenarkan melakukan kesilapan ketika membuat program.

Sistem ini juga menawarkan bantuan kepada pelajar sekiranya diminta di mana bantuannya adalah meliputi dalam perancangan program, pengekodan, pengujian dan 'debugging' kod. ELMPE mempunyai modul seperti *example selection* dan *structure editor* yang membantu dalam pemilihan contoh kod program disamping dapat membantu pelajar dalam mengesan kesilapan serta memberi cadangan penyelesaian





Rajah 2.3 Antaramuka sistem ELMPE

### 2.3.3 Kekurangan ELMPE

Setelah dikaji sistem ELMPE ini mempunyai beberapa kekurangan untuk digunakan bagi menyelesaikan permasalahan yang dinyatakan di dalam bab 1 :

#### ➤ Menggunakan sistem pengendalian Mac OS

Penggunaan sistem pengendalian Mac OS tidak digunakan dengan meluas di Malaysia termasuk di FSKTM, Universiti Malaya.

➤ **Bahasa pengaturcaraan yang diajar adalah LISP**

LISP merupakan sejenis bahasa pengaturcaraan simbolik yang digunakan dalam bidang Kepintaran Buatan di mana penggunaannya di Malaysia tidak meluas serta kurang diketahui.

➤ **Sistem didalam bahasa Inggeris**

Bahasa yang digunakan di dalam ELMPE ialah bahasa Inggeris berbanding dengan bahasa Melayu yang menjadi bahasa pengantar utama dalam pembelajaran di FSKTM.

## **2.4 SUMBANGAN DAN PERBEZAAN S2PC**

Sistem sedia ada yang telah dinyatakan iaitu ELMPE , merupakan sebuah sistem pengajaran serta pembelajaran yang serba lengkap .Pembangunannya yang dijalankan oleh University of Trier diketuai oleh Prof. Weber [1992] mempunyai perbezaan yang begitu ketara dengan S2PC.Kajian yang dilakukan dalam membangunkan sistem ini adalah bertaraf tinggi di mana ia mendapat bantuan dan dan daripada pelbagai pihak termasuk daripada kerajaan Jerman.Sistem ini yang boleh digunakan untuk membuat program LISP serta mempunyai rangkuman daripada bantuan untuk memilih contoh hinggalah kepada debugging ternyata melampaui apa yang ditawarkan oleh S2PC.



Walaupun bagaimanapun pembangunan S2PC masih mempunyai perbezaan dan sumbangan yang boleh diberikan. S2PC boleh dilihat sumbangannya dalam konteks pembelajaran di peringkat pengajian tinggi di Malaysia. Selain daripada sistem yang menggunakan Windows sebagai sistem pengendalian yang mana penggunaannya meluas di Malaysia, sistem ini adalah dibuat untuk pelajar mempelajari bahasa pengaturcaraan C di mana C boleh dianggap sebagai batu asas kepada pelajar-pelajar Sains Komputer. Sumbangan dan perbezaan S2PC dengan ELMPE juga boleh dilihat dari sudut penggunaan bahasa di mana S2PC menggunakan bahasa Melayu yang merupakan bahasa pengantar utama dalam pengajaran di FSKTM.



## **BAB 3                      METODOLOGI**

### **3.1      KAEDAH MENDAPATKAN PENYELESAIAN**

Dalam projek ini, terdapat beberapa kaedah yang digunakan bertujuan untuk mendapatkan penyelesaian bagi permasalahan yang telah dinyatakan sebelum ini. Terdapat empat kaedah utama yang telah digunakan dalam projek ini iaitu :

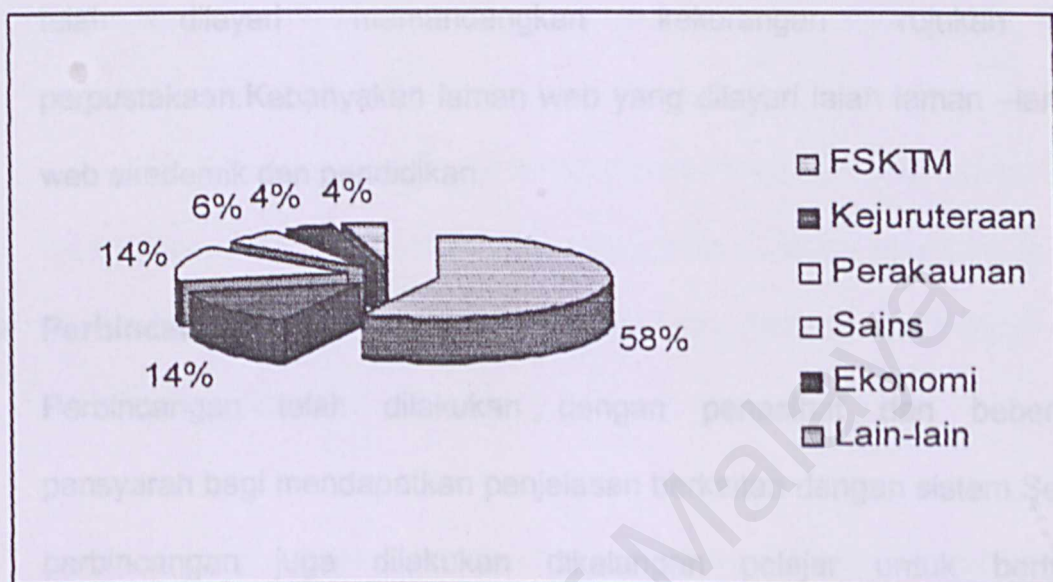
#### **➤ Bahan bacaan**

Bahan-bahan bacaan yang digunakan adalah seperti buku rujukan, jurnal, kertas kerja persidangan, projek ilmiah terdahulu, tesis serta nota-nota kuliah. Memandangkan CBR merupakan teknologi yang masih lagi baru di Malaysia maka sukar untuk mendapatkan bahan bacaan berbentuk buku. Justeru itu kajian ini lebih banyak menggunakan bahan bacaan berbentuk jurnal dan kertas kerja persidangan.

#### **➤ Soal-selidik**

Soal selidik adalah teknik pengumpulan fakta yang menggunakan borang atau dokumen tertentu bagi mendapatkan maklumbalas responden [16]. Di dalam projek ini soal selidik digunakan untuk mendapatkan maklumat mengenai kaedah yang digunakan oleh pelajar dalam pembelajaran sesuatu kursus. Kaedah persampelan yang digunakan ialah persampelan mudah memandangkan kaedah ini membolehkan borang

diperolehi kembali dengan cepat. Sebanyak 70 responden telah mengambil bahagian di mana peratusannya ditunjukkan dalam rajah 3.1.



Rajah 3.1: Peratusan agihan borang mengikut fakulti

Respondan utama soal-selidik ini ialah pelajar FSKTM di mana 40 orang responden telah mengambil bahagian. Baki responden adalah 10 orang masing-masing bagi Fakulti Kejuruteraan dan Perakaunan, 4 orang dari Fakulti Sains, 3 orang dari Fakulti Ekonomi dan masing-masing satu untuk pelajar dari Fakulti Akaedemi Pengajian Islam, Fakulti Alam Bina dan Fakulti Undang-undang.



### ➤ Internet

Beberapa laman yang berkaitan dengan CBR,C,Visual Basic dan Access telah dilayari memandangkan kekurangan rujukan di perpustakaan.Kebanyakan laman web yang dilayari ialah laman –laman web akademik dan pendidikan.

### ➤ Perbincangan

Perbincangan telah dilakukan dengan penasihat dan beberapa pensyarah bagi mendapatkan penjelasan berkaitan dengan sistem.Selain perbincangan juga dilakukan dikalangan pelajar untuk bertukar pandangan serta pendapat.

## 3.2 METODOLOGI PEMBANGUNAN SISTEM

Kitar Hayat Pembangunan Sistem (*System Development Life Cycle*) adalah metodologi pembangunan sistem yang telah dipiawaikan bagi memastikan proses pembangunannya mematuhi standard yang ditentukan. Jujukan langkah-langkah pembangunan yang teratur dikenali sebagai kitar hayat adalah bertujuan untuk memastikan proses pembangunan diketahui oleh mereka yang terlibat dalam projek berkenaan dan individu yang berminat terhadap pembangunan sistem.



### 3.2.1 Model Air Terjun dengan Prototaip

Pelbagai jenis model perisian yang menggunakan pendekatan Kitar Hayat Pembangunan Sistem telah diperkenalkan antaranya adalah model Air Terjun (*waterfall*), model V, model Prototaip, model spesifikasi beroperasi, model transformasi dan lain-lain. Bagi membangunkan sistem ini, model Air Terjun (*Waterfall*) dengan Prototaip telah digunakan sebagai metodologi panduan.

Model air terjun dengan prototaip merupakan ubahsuaian daripada model air terjun. Pembangunan sistem menggunakan model ini adalah dibuat berdasarkan kepada lima fasa iaitu:

- **Fasa Kajian Awal**

Fasa ini merupakan kajian awal bagi mendapatkan maklumat dan gambaran tentang sistem yang akan dibangunkan.

- **Fasa Analisa Sistem**

Fasa analisa sistem pula bertujuan untuk memahami bagaimana sistem yang akan dibangunkan dapat menyelesaikan masalah yang telah dikenalpasti dari kajian awal, ianya meliputi kajian

sistem, keperluan dan spesifikasi sistem, cadangan pengisian sistem, analisis alatan pembangunan sistem

➤ **Fasa Rekabentuk Sistem**

Dalam fasa ini semua ciri-ciri sistem seperti senibina sistem, rekabentuk pangkalan data, rekabentuk proses seperti carta struktur dan rekabentuk antaramuka pengguna diuraikan

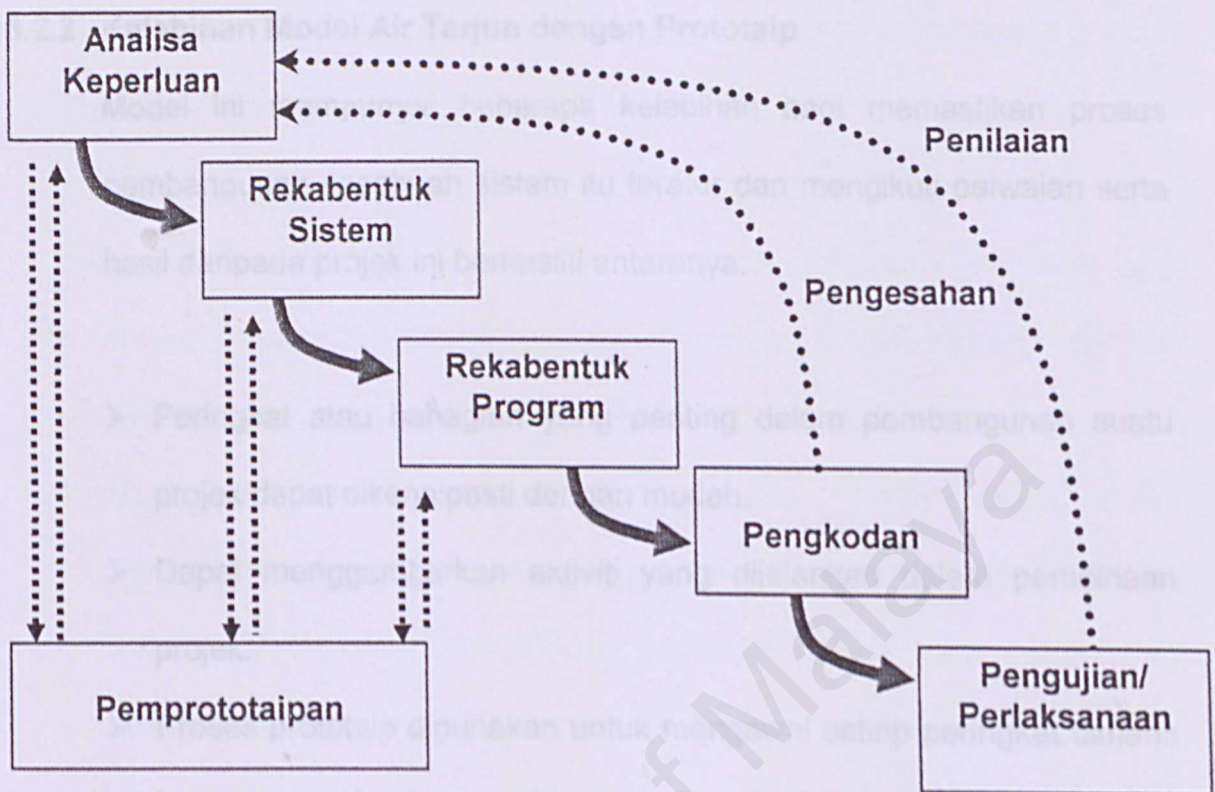
➤ **Fasa Pengkodan/Pembangunan**

Pada fasa ini aturcara program akan ditulis dan seterusnya dibangunkan. Fasa ini amat penting kerana ia akan menentukan sama ada sistem akan mencapai objektifnya.

➤ **Fasa Pengujian dan Perlaksanaan**

Proses untuk menguji keberkesanan sesuatu aturcara itu menjalankan fungsinya dan ianya bertujuan untuk mencari ralat pada sesuatu sistem itu dan menjejaki kesilapan aturcara.





Rajah 3.2: Model air terjun dengan pemprototaipan

Pemprototaipan ialah produk yang dibangunkan separuh yang membenarkan pelanggan dan pembangun untuk memeriksa dan menilai sebahagian dari aspek sistem yang dicadangkan. Penilaian bertujuan memastikan sistem telah melaksanakan semua keperluan, manakala pengesahan digunakan untuk memastikan setiap fungsi berjalan dengan betul.



### 3.2.2 Kelebihan Model Air Terjun dengan Prototaip

Model ini mempunyai beberapa kelebihan bagi memastikan proses pembangunan sesebuah sistem itu teratur dan mengikuti pawai serta hasil daripada projek ini berkualiti antaranya:

- Peringkat atau bahagian yang penting dalam pembangunan suatu projek dapat dikenalpasti dengan mudah.
- Dapat menggambarkan aktiviti yang dijalankan dalam pembinaan projek.
- Proses prototaip digunakan untuk mengawal setiap peringkat dimana ia dapat membantu menafsir strategi-strategi rekabentuk yang lain.
- Penggunaan kaedah penilaian (*validation*) bagi memastikan sistem mengimplementasikan semua keperluan, supaya setiap fungsi sistem boleh dijejak ke keperluan tertentu dalam spesifikasi.
- Penggunaan ujian pengesahan (*verification*) keperluan bagi memastikan setiap fungsi berjalan dengan betul.

### 3.3 KEPERLUAN PERISIAN

Penentuan jenis perisian yang akan digunakan dalam pembangunan sistem adalah sangat penting supaya sistem yang akan dibangunkan nanti bersesuaian dengan konsep dan skop yang disasarkan. Perisian yang telah dipilih untuk

membangunkan S2PC ialah Visual Basic 6.0. Antara sebab mengapa Visual Basic 6.0 dipilih ialah :

- Visual Basic menggabungkan satu set teknologi perisian yang dipanggil Active X yang membenarkan ciptaan, integrasi, penggunaan semula komponen perisian yang dipanggil kawalan
- Mempunyai ciri-ciri antaramuka pengguna yang menarik serta memudahkan pengguna sebagai contoh *tabstrip* yang diguna dalam S2PC membuatkan borang(*form*) pemilihan contoh lebih kemas dan tersusun. Di samping itu ia turut menawarkan grafik seperti ikon yang menarik dan memudahkan pemahaman pengguna
- Membolehkan pengubahan ciri-ciri pada borang atau sub borang secara program
- Perisian ini juga dapat bergabung dengan Microsoft Access 2000 sebagai pangkalan datanya
- Membolehkan sistem dibina dalam bentuk stand alone dan memudahkan lagi proses menjadikan sistem dalam bentuk *execute*.

Pembinaan pangkalan pengetahuan di dalam S2PC pula dibina dengan menggunakan Microsoft Access 2000. Antara kelebihan Access ialah:

- Mudah untuk memasukkan data ke dalam pangkalan pengetahuan iaitu dengan penggunaan *table*



- Dapat diintegrasikan dengan Visual Basic dengan mudah iaitu dengan penggunaan fungsi ADO DC yang terdapat di dalam Visual Basic

Sistem pengendalian yang digunakan untuk membangunkan sistem ini pula ialah Windows 98 memandangkan penggunaannya yang meluas dan lebih dikenali ramai berbanding sistem pengendalian yang lain.

### 3.4 KEPERLUAN PERKAKASAN

Perkakasan biasanya merujuk kepada mesin atau peralatan fizikal yang melakukan fungsi-fungsi asas yang terkandung di dalam kitaran pengoperasian proses. Spesifikasi minimum yang diperlukan untuk melarikan Sistem Pemilihan Contoh Pengaturcaraan C adalah seperti berikut:

- Pemprosesan Intel Pentium MMX
- 32+ RAM memori
- 500 MB ruangan cakera keras minimum
- Monitor 256-warna paparan SVGA (resolusi 800\*600)

## BAB 4

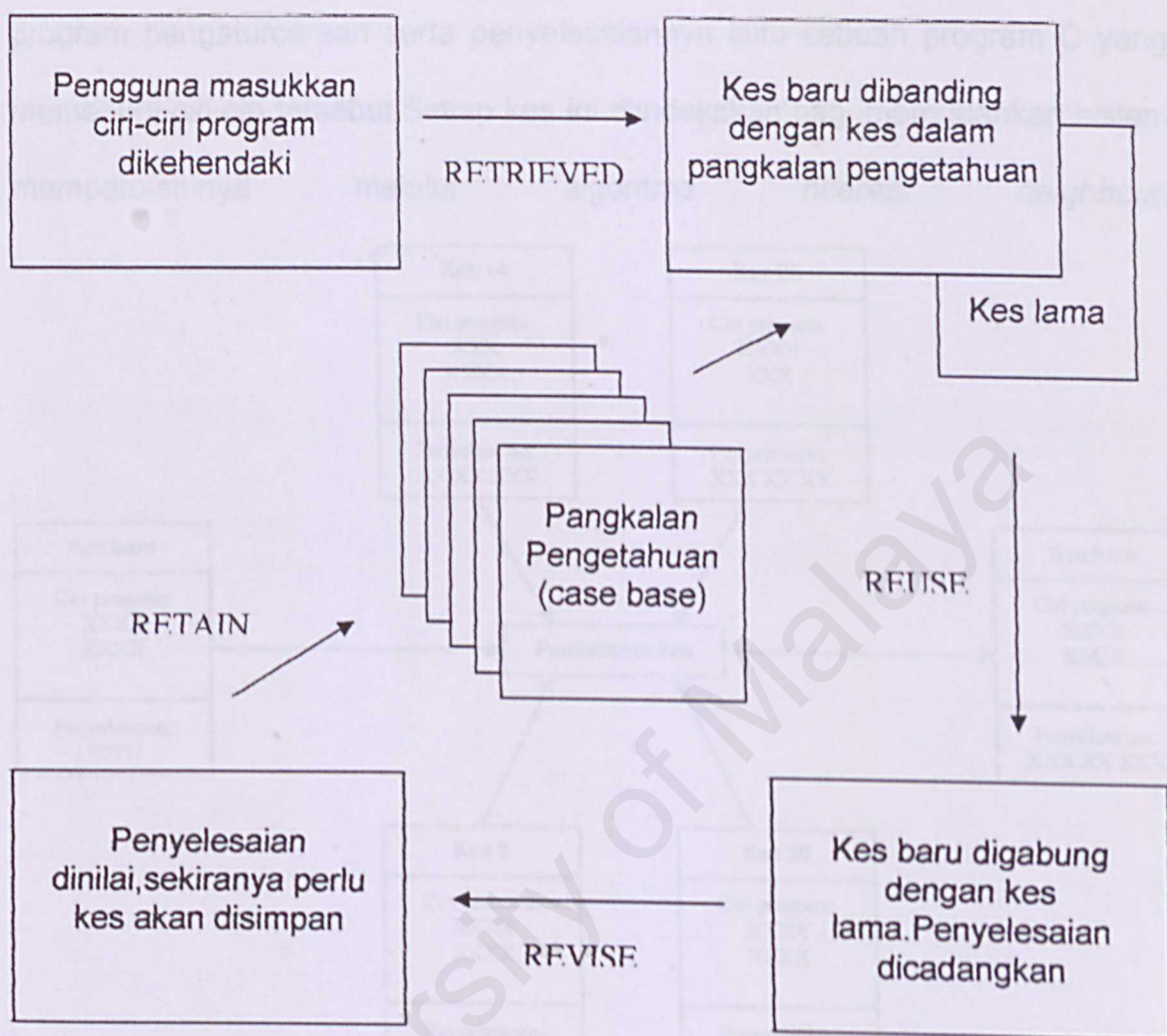
## REKABENTUK SISTEM

### 4.1 REKABENTUK S2PC

Rekabentuk merupakan satu proses kreatif yang menukarkan masalah kepada penyelesaian [17]. Fasa ini merangkumi semua tugas dan fungsi yang memberi keutamaan kepada spesifikasi terperinci dan mendalam berasaskan penyelesaian masalah [17]. Ia digambarkan sebagai satu proses yang menjadi perwakilan kepada struktur data, struktur program, ciri-ciri antaramuka dan maklumat-maklumat berprosedur. Segala maklumat yang dikumpulkan semasa fasa analisa sistem akan diubah dalam bentuk modul-modul yang kecil sehinggalah terbentuknya sebuah sistem.

S2PC merupakan sebuah sistem yang menggunakan kaedah CBR maka rekabentuk sistem ini adalah menyamai kitar CBR. Rajah 4.1 menunjukkan bagaimana sistem SPCPP berfungsi. Berdasarkan rekabentuk tersebut ciri-ciri contoh program C yang ingin dirujuk oleh pengguna dinyatakan sebagai masalah ataupun kes yang baru. Sistem, melalui fasa *retrieve* akan merujuk pangkalan pengetahuan dan memadankan kes baru tersebut dengan kes di dalam pangkalan pengetahuan. Seterusnya sistem akan mencadangkan contoh program yang sesuai dengan mengabungkan huraian di dalam kes baru dengan kes lama semasa fasa *reuse*. Fasa *revision* didalam sistem pula akan menilaikan kebolehsuaian program C yang telah dicadangkan.



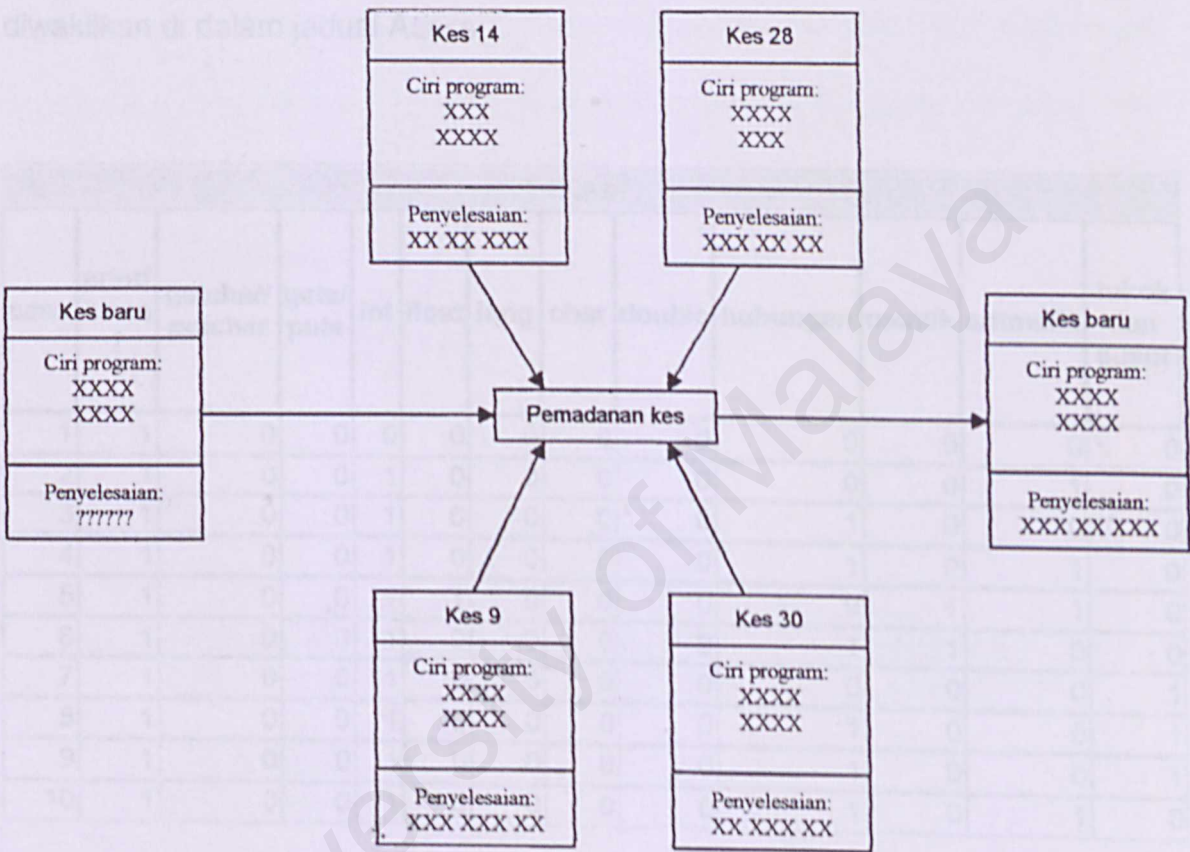


Rajah 4.1: Rekabentuk sistem S2PC = Kitar CBR

## 4.2 REKABENTUK PANGKALAN DATA

Pangkalan data di dalam CBR adalah dirujuk sebagai pangkalan pengetahuan yang mengandungi penyelesaian kes-kes lalu ataupun dinyatakan juga sebagai *case base*. Kes-kes di dalam sistem ini merupakan ciri-ciri bagi sesebuah

program pengaturcaraan serta penyelesaiannya iaitu sebuah program C yang memenuhi ciri-ciri tersebut. Setiap kes ini diindekskan bagi memudahkan sistem memperolehnya melalui algoritma *nearest neighbour*.



Rajah 4.2: Ciri-ciri program (kes) di dalam pangkalan pengetahuan[Choy & Lee]

Di dalam pembinaan pangkalan pengetahuan S2PC, perisian Microsoft Access 2000 telah digunakan di mana ciri-ciri bagi bahasa C bagi setiap kes disusun dalam bentuk jadual. S2PC telah menyenaraikan sebanyak 25 ciri yang terdapat dalam topik-topik asas pengaturcaraan C serta 100 kes iaitu contoh-contoh pengaturcaraan C. Bagi setiap kes, ciri-ciri dalam bahasa C yang terdapat di



dalamnya akan diwakili dengan nilai 1 di dalam jadual(*table*) Access manakala ciri-ciri yang tidak terdapat dalam kes tersebut akan diberikan nilai 0. Jadual 4.1 menunjukkan contoh bagaimana 10 kes dan 12 daripada 25 ciri bahasa C diwakilkan di dalam jadual Access.

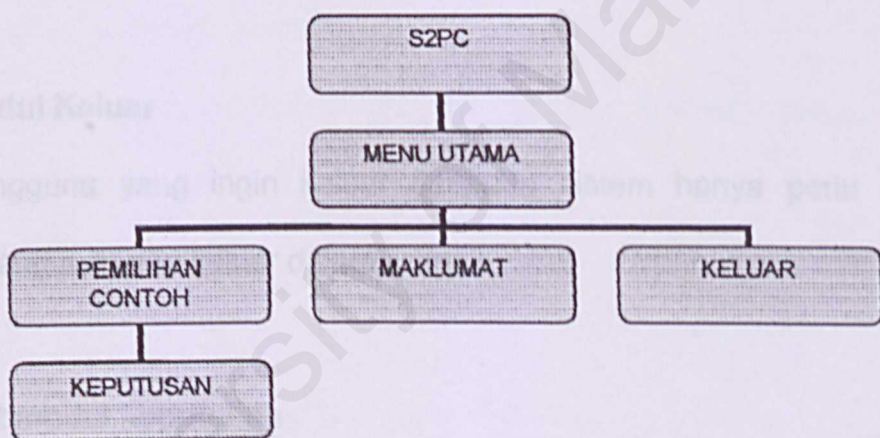
Kes												
case	printf/scanf	getchar/putchar	gets/puts	int	float	long	char	double	hubungan	mantik	aritmetik	tokok dan susut
1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	1	0
3	1	0	0	1	0	0	0	0	1	0	0	0
4	1	0	0	1	0	0	0	0	1	0	1	0
5	1	0	0	1	1	0	0	0	0	1	1	0
6	1	0	0	1	0	0	0	0	1	1	0	0
7	1	0	0	1	0	0	0	0	0	0	0	1
8	1	0	0	1	0	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0	1	0	0	1
10	1	0	0	1	0	0	0	0	1	0	1	0

Jadual 4.1:Perwakilan sebahagian kes dan ciri di dalam pangkalan data

Penyelesaian kes iaitu aturcara C bagi setiap kes pula dimasukkan di dalam sebuah lagi jadual di dalam Microsoft Access. Di dalam jadual ini setiap penyelesaian iaitu aturcara C bagi setiap kes dimasukkan berpandukan nombor kesnya.

### 4.3 REKABENTUK STRUKTUR

Untuk memberi gambaran mengenai struktur Sistem Pemilihan Contoh Program C, carta struktur digunakan. Merujuk kepada carta yang berikut, S2PC telah dibahagikan kepada komponen-komponen mengikut fungsinya. Komponen-komponen yang berada di paras atas akan mewakili fungsi-fungsi yang akan ditemui di paparan awal antaramuka sistem dan diikuti dengan komponen-komponen seterusnya.



Rajah 4.3: Carta Struktur Sistem Pemilihan Contoh Pengaturcaraan C

#### 4.3.1 Modul yang terlibat

Berdasarkan carta struktur yang dinyatakan dapat dilihat S2PC mempunyai empat modul utama dan satu submodul.



#### ➤ Modul Pemilihan Contoh

Modul ini adalah modul paling penting di dalam sistem ini, kerana melalui modul inilah pelajar akan memasukkan ciri-ciri dalam program C yang dikehendaki untuk membolehkan sistem mencadangkan contoh program yang sesuai dijadikan rujukan

#### ➤ Modul Maklumat

Modul ini hanya melibatkan pemberitahuan umum mengenai sistem ini

#### ➤ Modul Keluar

Pengguna yang ingin keluar daripada sistem hanya perlu menekan butang fungsian keluar dalam sistem.

#### ➤ Submodul keputusan

Submodul keputusan akan aktif apabila pengguna melakukan pemilihan pada modul pemilihan contoh. Di dalam modul inilah contoh program C yang dipilih serta penerangan mengenainya akan dipaparkan.

#### 4.4.2 Keperluan kefungsian

Keperluan kefungsian menerangkan interaksi di antara sistem dan juga persekitarannya. Keperluan kefungsian yang terdapat di dalam Sistem Pemilihan Contoh Pengaturcaraan C ialah :

➤ **Fungsi Pemilihan**

Fungsi ini akan menerima input daripada pengguna yang merupakan ciri-ciri sesuatu program. Proses pepadanan tidak akan dijalankan sekiranya pengguna tidak memasukkan input.

➤ **Fungsi Keluar**

Pengguna boleh keluar daripada program pada bila-bila masa sahaja

➤ **Fungsi Pemilihan Semula**

Pemilihan semula contoh program boleh dilakukan semula apabila butang fungsi ini diaktifkan.

#### 4.4.3 Keperluan bukan fungsian

Keperluan bukan fungsian adalah deskripsi bagi ciri-ciri yang menyempurnakan lagi sesuatu sistem dan juga kekangan-kekangan yang menghadkan skop [16]. Keperluan bukan fungsian dalam sistem ini ialah :

➤ **Mesra pengguna**

Sistem adalah mudah digunakan dan berinteraksi dengan pengguna



➤ **Keselamatan**

S2PC selamat digunakan tanpa memberi gangguan pada sistem komputer pengguna.

➤ **Kebolehpercayaan**

Sistem tidak akan mengalami kerosakan yang teruk sekiranya terdapat masalah yang tidak dapat dielakkan.

➤ **Keringkasan**

Skrin dan arahan diatur dengan tersusun bagi memudahkan pemahaman pengguna dan boleh menarik minat pengguna

➤ **Masa tindakbalas**

Urusan dan tindakbalas yang berlaku dalam sistem mestilah dapat dilakukan dalam jangka masa yang munasabah

#### 4.4 REKABENTUK ANTARAMUKA

Pengguna (manusia) perlu berkomunikasi dengan sistem yang dibangunkan untuk menjalankan urusannya. Oleh itu pengguna memerlukan medium khusus untuk berkomunikasi dengan sistem iaitu antaramuka pengguna. Sekiranya

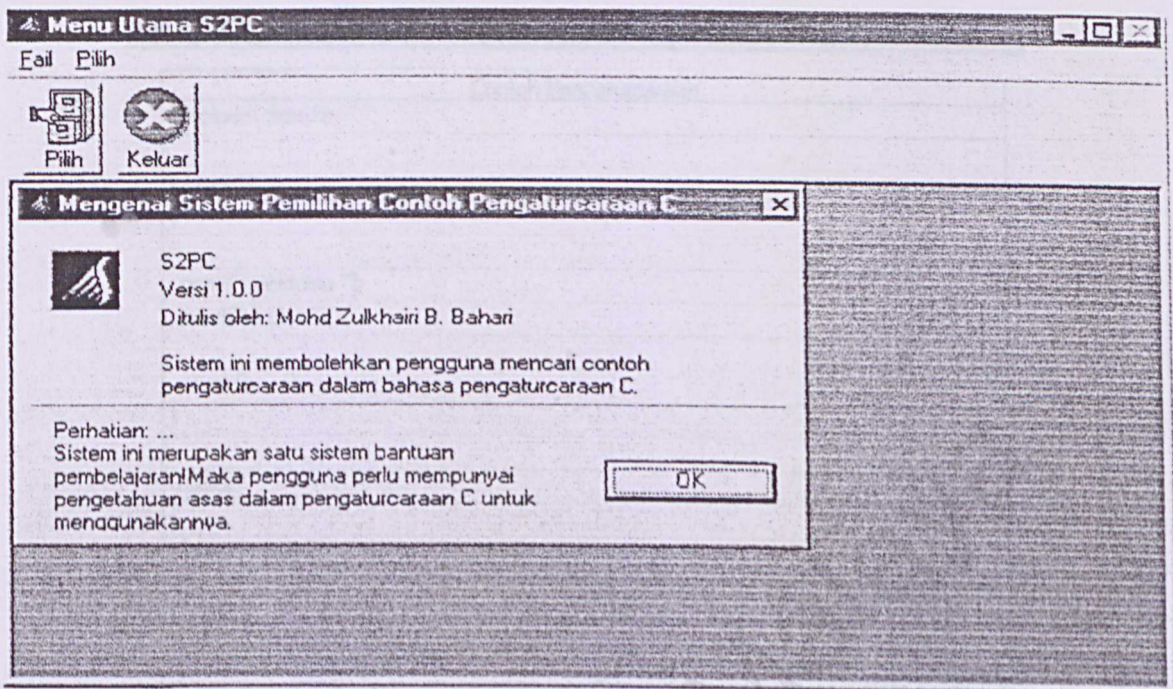
antaramuka adalah baik, ia mampu menjadi penghubung yang baik di mana segala maklumat yang diperlukan oleh kedua-dua belah pihak dapat disampaikan dengan sempurna.

Rekabentuk antaramuka pengguna adalah amat penting dalam pembangunan sesebuah perisian kerana rekabentuk yang menarik akan menambahkan lagi minat pengguna untuk mencuba sesuatu perisian itu. Antaramuka pengguna secara amnya adalah :

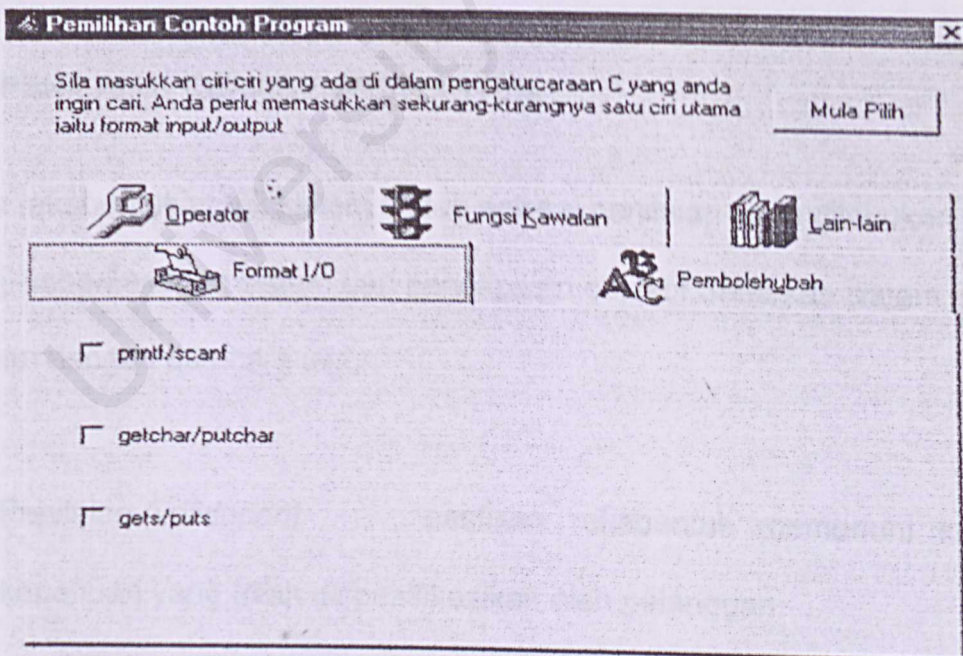
- Spesifikasi dialog atau perhubungan antara manusia dengan komputer
- Dialog adalah dalam bentuk input data dan output maklumat
- Pengguna masukkan data yang dikehendaki ke dalam sistem melalui antaramuka untuk diproses
- Selepas diproses, sistem akan mengeluarkan output yang boleh dicapai oleh pengguna melalui antaramuka

Rajah 4.4 ,4.5 dan 4.6 menunjukkan antaramuka pengguna yang digunakan di dalam S2PC.

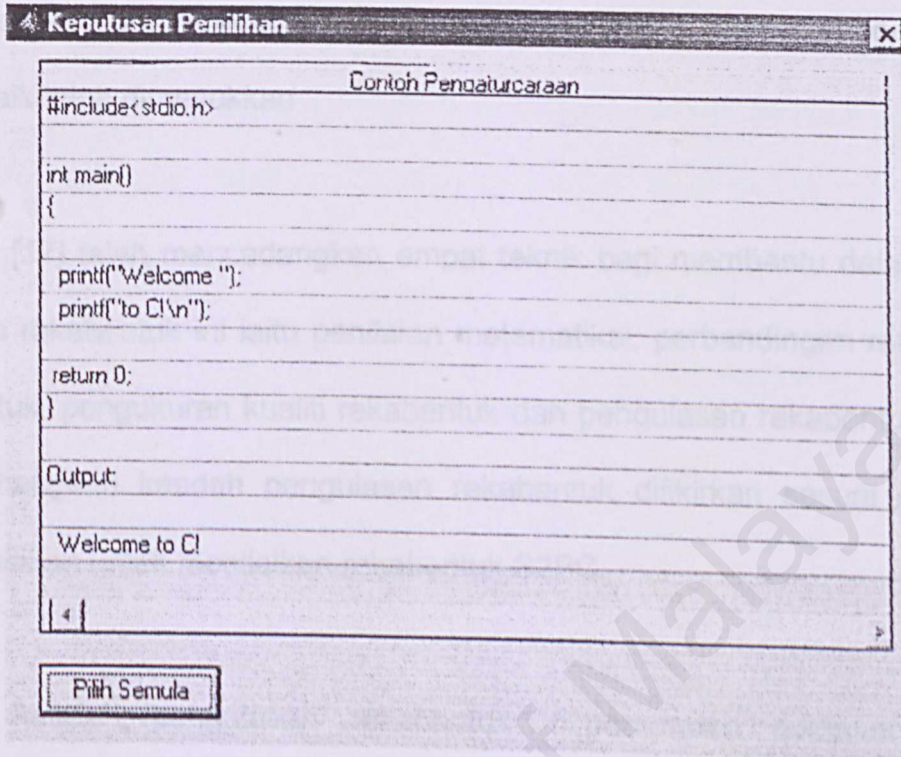




Rajah 4.4 Antaramuka untuk menu utama S2PC



Rajah 4.5: Antaramuka untuk modul pemilihan



Rajah 4.6: Antaramuka keputusan pemilihan

#### 4.5 PENILAIAN REKABENTUK SISTEM

Apabila rekabentuk untuk sistem sudah selesai, penilaian perlu dilakukan untuk menguji keberkesanan sistem dan pencapaian objektif. Semakan sistem ini dilakukan dengan dua cara iaitu:

- a) Penilaian (*validation*) : pastikan rekabentuk memenuhi kesemua keperluan yang telah dispesifikasikan oleh pelanggan



b) Pengesahan(*verification*) : memastikan semua ciri-ciri rekabentuk yang baik telah dimasukkan

Pfleeger [17] telah mencadangkan empat teknik bagi membantu dalam proses penilaian rekabentuk ini iaitu penilaian matematikal, perbandingan rekabentuk-rekabentuk, pengukuran kualiti rekabentuk dan pengulasan rekabentuk. Setelah dipertimbangkan kaedah pengulasan rekabentuk difikirkan sesuai dan telah diadaptasikan untuk menilaikan rekabentuk S2PC.

Melalui kaedah pengulasan rekabentuk ini, pertemuan diadakan dengan pengguna dan beberapa orang penting untuk mendapat ulasan sebelum pembangunan sistem diteruskan. Dalam konteks sistem ini, orang penting yang dimaksudkan adalah merujuk kepada penasihat, moderator serta pembangun-pembangun sistem yang berminat terhadap projek ini.

Proses pengulasan ini dilakukan dalam tiga langkah iaitu pengulasan rekabentuk awal (*preliminary design review*), pengulasan rekabentuk kritikal (*critical design review*) dan pengulasan rekabentuk program (*program design review*). Pada mulanya pengulasan rekabentuk awal dilakukan untuk menguji rekabentuk dengan pengguna. Setelah pengguna berpuas hati pengulasan rekabentuk kritikal akan dilakukan bersama dengan penasihat, moderator atau pembangun sistem yang berminat di mana pemeriksaan teknikal secara

terperinci akan dilakukan. Akhir sekali, pengulasan rekabentuk program dilaksanakan di mana pereka program akan menafsirkan huraian rekabentuk untuk pengkodan dan pengujian. Setelah itu, hasilnya akan dibentangkan kepada penasihat, moderator atau pembangun sistem yang berminat bagi mendapatkan komen serta cadangan.

5.1 PENGKODAN DAN IMPLEMENTASI S2FC

Pembangunan sistem, fasa pengkodan dan implementasikan sistem ini dinyatakan sebagai amat penting kerana ia akan menentukan sama ada sistem mencapai objektifnya. Dalam fasa ini skema program ditulis dan semuanya dibangunkan dengan mengimplementasikan data dan serta keperluan pensian yang telah dinyatakan dalam fasa-fasa sebelumnya dalam pembangunan sistem sebelum ini. Pengkodan dan implementasi merupakan fasa yang memakan masa yang agak lama memandangkan ia memerlukan ketelitian yang tinggi supaya tidak tersasar daripada tujuan pembangunan sistem. Pembangunan Sistem Pemilihan Corak dalam C ini dilakukan dengan menggunakan perisian Visual Basic 6.0 serta penggunaan Microsoft Access 2000 sebagai pangkalan data.

### 5.1.1 Kandungan dan Antaramuka S2FC

Kandungan sesuatu sistem atau apa yang perlu diperlihatkan kepada pengguna dan operasi yang boleh digunakan oleh pengguna adalah sangat penting dalam fasa ini kerana ia perlu memenuhi keperluan sistem (system requirement) serta rekabentuk sistem. Ia juga harus mengandungi keseluruhan ciri-ciri dan fungsi yang diperlukan oleh pengguna secara umum.



## 5.1 PENGKODAN DAN IMPLEMENTASI S2PC

Sebagaimana yang dijelaskan dalam bab 3 iaitu metodologi, di dalam kitar hayat pembangunan sistem, fasa pengkodan dan implementasikan sistem ini dinyatakan sebagai amat penting kerana ia akan menentukan sama ada sistem mencapai objektifnya. Dalam fasa ini aturcara program akan ditulis dan seterusnya dibangunkan dengan mengimplementasikan kaedah serta keperluan perisian yang telah dinyatakan dalam fasa-fasa kitar hayat pembangunan sistem sebelum ini. Pengkodan dan implelementasi ini merupakan fasa yang memakan masa yang agak lama memandangkan ia memerlukan ketelitian yang tinggi supaya tidak tersasar daripada objektif pembangunan sistem. Pembangunan Sistem Pemilihan Contoh Program C ini dilakukan dengan menggunakan perisian Visual Basic 6.0 beserta penggunaan Microsoft Access 2000 sebagai pangkalan pengetahuan.

### 5.1.1 Kandungan dan Antaramuka S2PC

Kandungan sesuatu sistem iaitu apa yang perlu dipaparkan kepada pengguna dan operasi yang boleh digunakan oleh pengguna adalah sangat penting dalam fasa ini kerana ia perlu memenuhi keperluan sistem (*system requirement*) serta rekabentuk sistem. Ia juga harus mengandungi kesemua ciri-ciri dan fungsi yang diperlukan oleh pengguna secara umum.

### 5.1.2 Pengintegrasian pangkalan pengelakan

Menu utama bagi S2PC adalah dibina dengan penggunaan satu borang (*form*) utama iaitu borang *Multiple Document Interface (MDI)* dan mempunyai empat sub borang (*MDI child*) yang mewakili setiap modul yang lain, yang disediakan di dalam perisian Visual Basic 6.0.

Visual Basic 6.0 menawarkan beberapa ciri yang terdapat dalam borang utama ini iaitu penggunaan aplikasi *toolbar* dan *menu editor* sebagai medium untuk membuka sub-sub modul yang lain serta penggunaan *icon* dan kunci jalan pintas (*short cut key*). Penggunaan kedua-dua ciri ini bukan sahaja memberikan rekabentuk antaramuka yang baik malahan memudahkan pengguna sistem dalam memahami dan menggunakan sistem ini.

Sebagaimana yang telah digambarkan di dalam bab 4 iaitu rekabentuk sistem, S2PC menggunakan butang-butang yang bersesuaian dengan keperluan seperti *command button* dan *check button*. Bagi pemilihan ciri-ciri untuk pengaturcaraan C pula, S2PC menggunakan satu fungsi yang disediakan dalam Visual Basic iaitu *tabstrip* seperti yang ditunjukkan dalam rajah . Fungsi ini memudahkan pengguna untuk memilih ciri-ciri yang dikehendaki kerana ia membolehkan setiap kategori bagi ciri-ciri tersebut disusun dengan kemas dan teratur. Setiap kategori tersebut disusun dalam bentuk tatasusunan (*array*) oleh itu hanya kategori yang mahu dipilih ciri-cirinya sahaja akan dipaparkan.



### 5.1.2 Pengintegrasian pangkalan pengetahuan

Pengintegrasian sistem dengan pangkalan pengetahuan adalah penting kerana melalui pengintegrasian inilah data yang terdapat di dalam pangkalan pengetahuan dapat dicapai. S2PC menggunakan fungsi ADODC yang terdapat di dalam Visual Basic untuk mengintegrasikan data yang terdapat di dalam jadual Microsoft Access dengan kandungan sistem S2PC. Pengintegrasian ini disambungkan dengan menggunakan *provider* Microsoft.Jet.OLEDB.4.0 serta penggunaan kod SQL (structured query language).

### 5.1.3 Pengkodan case- scoring

Fasa pengkodan ini adalah fasa yang penting sekali dalam bahagian pengkodan Sistem Pemilihan Contoh Pengaturcaraan C ini, kerana pada fasa inilah *case-based reasoning* diimplementasikan ke dalam sistem. Dalam sistem pemilihan contoh dilakukan secara *case-scoring* yang dilakukan berpandukan kepada algoritma *nearest neighbour*.

Melalui algoritma *nearest neighbour* ini ciri-ciri bagi setiap kes diberikan nilai pemberat. Seterusnya nilai-nilai pemberat bagi setiap ciri ditambahkan kesemuanya dan kes yang mempunyai hasil tambah ataupun *scoring* yang paling tinggi akan dipilih sebagai penyelesaian.

S2PC telah menyenaraikan sebanyak 25 ciri-ciri yang boleh dipilih oleh pengguna untuk memilih contoh pengaturcaraan C [ rujuk lampiran ].Setiap ciri-ciri ini boleh dipilih dengan menggunakan *checkbox*.Melalui *checkbox* tersebut ,sekiranya sesuatu ciri dipilih maka ia akan bernilai satu dan sebaliknya bernilai kosong apabila tidak dipilih.

*Case scoring* dilakukan dengan membandingkan nilai-nilai ciri setiap kes yang terdapat di dalam pangkalan pengetahuan [ Rujuk Jadual 5.1 ] dengan nilai yang dibawa oleh *checkbox* yang mewakili sesuatu ciri dalam pengaturcaraan C.Setiap ciri yang bernilai 1 yang dibawa oleh *checkbox* mempunyai keutamaan yang tinggi di mana sekiranya ia berpadanan dengan ciri kes di dalam pangkalan pengetahuan maka pemberatnya akan bernilai 3.Manakala jika nilai 0 berpadanan dengan pangkalan pengetahuan maka pemberatnya adalah bernilai 1.

Nilai dibawa <i>checkbox</i>	Nilai ciri di dalam pangkalan pengetahuan	Nilai pemberat
1	1	3
1	0	0
0	1	0
0	0	1

Jadual 5.1 : Jadual pemadanan nilai untuk dilakukan case-scoring



Contoh pengkodan bagaimana case-scoring dilakukan untuk beberapa ciri pengaturcaraan C ditunjukkan di bawah.

```
Select Case .Fields("printf/scanf").Value
```

```
Case 1      If printf.Value = 1 Then
              Else  tempscore = tempscore + 3
              tempscore = tempscore + 0
            End If
```

```
Case 0      If printf.Value = 0 Then
              tempscore = tempscore + 1
            Else
              tempscore = tempscore + 0
            End If
          End Select
```

```
Select Case .Fields("getchar/putchar").Value
```

```
Case 1      If getchar.Value = 1 Then
              Else  tempscore = tempscore + 3
              tempscore = tempscore + 0
            End If
```

Case 0

If getchar.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("gets/puts").Value

Case 1

If gets.Value = 1 Then

Else temp score = temp score + 3

temp score = temp score + 0

End If

Case 0

If gets.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select



## 5.2 RUMUSAN

Daripada penerangan bab 5 ini, dapat disimpulkan bahawa fasa pengkodan adalah fasa yang amat rumit dan kompleks di dalam proses pembangunan sesuatu sistem. Proses pengkodan ini penting dalam memastikan S2PC mencapai objektifnya menggunakan CBR dalam penyelesaian masalahnya di samping penghasilan antaramuka-antaramuka pengguna yang ramah pengguna yang membolehkan pengguna sistem dapat menggunakan sistem ini dengan lebih mudah.

### 5.1.1 Jarak-Jarak Ralat dan Kegagalan [17]

Dalam fasa pengujian ini, ujian yang baik ialah ujian yang mampu mengesan ralat-ralat yang boleh dapat di kesan bersama-sama analisa, rekabentuk dan pengkodan. Sesuatu pengujian dianggap berjaya hanya apabila ia menemui

## 6.1 TUJUAN PENGUJIAN

Pengujian perlu dilakukan untuk menjamin kualiti sesuatu perisian atau sistem yang dibangunkan. Selain itu pengujian sangat penting kerana ia memastikan sistem melaksanakan kehendak pengguna dan mencapai objektif pembangunan. Penentuan pencapaian objektif sistem ini melibatkan proses penelitian semula spesifikasi-spesifikasi, rekabentuk dan pengkodan yang telah dijalankan sepanjang membangunkan sistem.

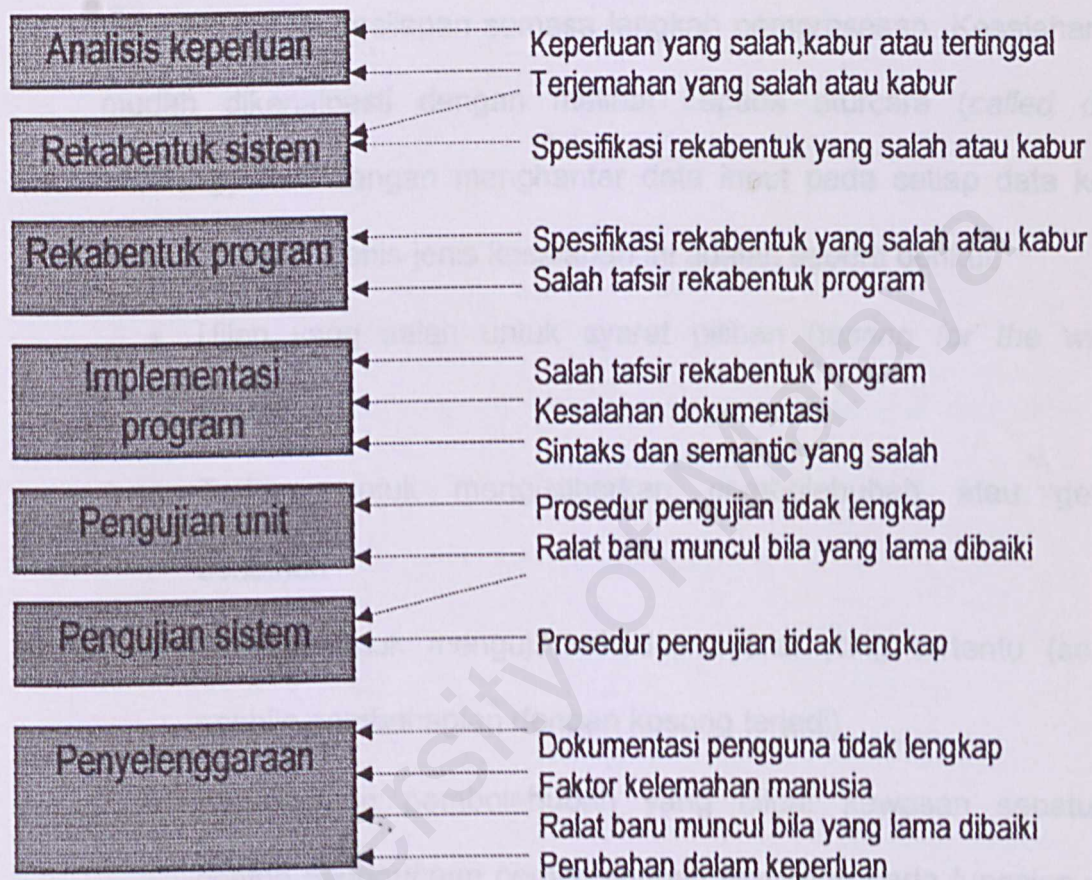
Pengujian juga dibuat bagi memastikan modul-modul yang dibina bebas daripada sebarang ralat yang boleh menyebabkan ketidakbolehpercayaan pada sistem dan melakukan apa yang sepatutnya dilakukan serta menghasilkan apa yang sepatutnya. Pengujian ini dilakukan dengan menggunakan data-data percubaan dan logik-logik yang digunakan di dalam pengkodan.

### 6.1.1 Jenis-jenis Ralat dan Kegagalan [17]

Dalam fasa pengujian ini ujian yang baik ialah ujian yang mampu mengenalpasti ralat-ralat yang tidak dapat dikesan semasa fasa analisis, rekabentuk dan pengkodan. Sesuatu pengujian dianggap berjaya hanya apabila kita menemui



ralat atau kegagalan. Rajah 6.1 menunjukkan punca-punca yang membolehkan wujudnya ralat semasa pembangunan sistem.



Rajah 6.1: Punca-punca ralat semasa pembangunan [Pfleger, 2001]

Terdapat pelbagai jenis ralat atau kegagalan yang boleh wujud dalam pembangunan sesebuah sistem. Ralat-ralat ini boleh dikategorikan kepada beberapa jenis iaitu:

### ➤ Kesalahan Algoritma (*Algorithmic faults*)

Berlaku apabila komponen algoritma atau logik tidak menghasilkan output yang dikehendaki untuk input yang telah diberikan. Ini berlaku kerana sesuatu kesilapan semasa langkah pemprosesan. Kesalahan ini mudah dikenalpasti dengan melihat kepada aturcara (*called desk checking*) atau dengan menghantar data input pada setiap data kelas yang berlainan. Jenis-jenis kesalahan ini adalah seperti berikut:

- Ujian yang salah untuk syarat pilihan (*testing for the wrong condition*)
- Terlupa untuk mengistiharkan pembolehubah atau gelung berlainan
- Terlupa untuk menguji terhadap syarat yang tertentu (seperti apabila pembahagian dengan kosong terjadi)
- Penggunaan pembolehubah yang diluar kawasan sepatutnya contoh penggunaan pembolehubah tempatan pada fungsian yang lain.

### ➤ Kesalahan pengiraan dan ketepatan (*Computation & precision faults*)

Berlaku bila implementasi formula adalah salah ataupun tidak tepat dan tidak menghasilkan keputusan mengikut tahap ketepatan yang dikehendaki.



### ➤ Kesalahan Sintak (*Syntax faults*)

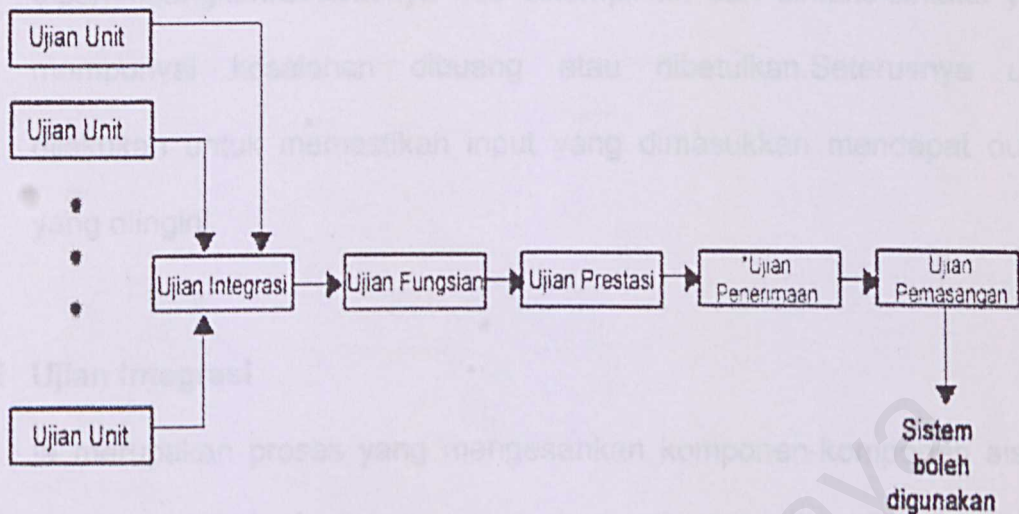
Kesalahan atau kesilapan sintak boleh diperiksa semasa berlakunya kesilapan algoritma. Selepas sesuatu program dijalankan jika berlaku kesalahan sintak ia akan dapat dikenalpasti dan memaklumkan jenis kesilapan serta dimana lokasinya.

### ➤ Kesilapan Dokumentasi (*Documentation faults*)

Berlaku apabila dokumentasi tidak sepadan dengan apa yang dilakukan oleh program. Kesilapan dokumentasi ini mungkin akan menyebabkan berlakunya ralat-ralat yang lain dalam program.

## 6.2 LANGKAH-LANGKAH PENGUJIAN

Ujian yang dibuat terhadap S2PC merupakan ujian secara menaik (*bottom-up testing*) iaitu yang bermula daripada unit-unit terkecil sehinggalah kepada pengujian sistem secara keseluruhan dan pemasangan sistem tersebut. Terdapat 6 ujian utama yang telah dijalankan dan turutan langkah-langkah pengujian ini adalah seperti yang ditunjukkan di dalam Rajah 6.2.



Rajah 6.2 Langkah-langkah Pengujian

### 6.2.1 Ujian Unit

Pengujian unit melibatkan pengujian ke atas unit-unit terkecil yang dikenali sebagai modul yang memfokuskan terhadap ketepatan, logik, syarat sempadan dan pengurusan ralat. Ia memastikan aliran maklumat yang tepat iaitu unit-unit dapat menerima input yang dimasukkan dan menghasilkan output seperti yang dijangkakan.

Pengujian unit dilakukan dengan mengkaji kod program keseluruhannya dan cuba mengenalpasti kesilapan algoritma, data dan sintaks. Kod-kod tersebut juga dibandingkan dengan spesifikasi keperluan dan juga rekabentuk untuk memastikan setiap kes yang relevan telah



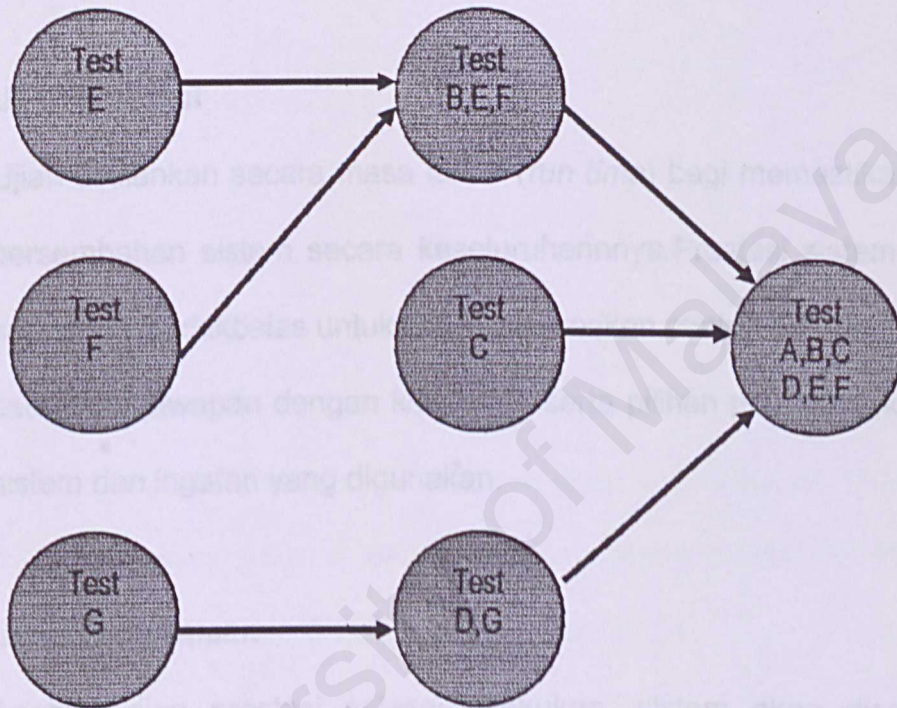
dipertimbangkan. Seterusnya kod dikompilkan dan sintaks-sintaks yang mempunyai kesalahan dibuang atau dibetulkan. Seterusnya ujian dilakukan untuk memastikan input yang dimasukkan mendapat output yang diinginkan.

### 6.2.2 Ujian Integrasi

Ia merupakan proses yang mengesahkan komponen-komponen sistem agar dapat bekerja bersama-sama seperti yang diuraikan di dalam spesifikasi rekabentuk program dan sistem. Ia memfokuskan kepada ujian satu atau lebih modul yang disepadukan. Di antara ujian integrasi yang dilaksanakan ialah memastikan borang (*form*) pemilihan dengan borang keputusan diintegrasikan. Data-data yang diinputkan ke dalam borang pemilihan dapat mencapai pangkalan pengetahuan dan seterusnya penghantaran data dari borang pemilihan dapat diterima di dalam borang keputusan.

Di dalam menjalankan ujian integrasi bagi projek ini, strategi yang telah digunakan ialah ujian integrasi bawah atas (*bottom-up*). Di dalam strategi ini, setiap komponen yang wujud di tahap yang paling rendah di dalam hierarki sistem akan diuji secara individual terlebih dahulu. Kemudian, komponen seterusnya yang perlu diuji adalah komponen yang mana

memanggil komponen yang telah diuji sebelumnya. Pendekatan ini dituruti sehingga semua komponen yang terlibat diuji.



Rajah 6.3 : Pengujian bawah atas [Pfleeger,2001]

### 6.2.3 Ujian Fungsi

Ujian fungsi pula akan menilai sistem untuk menentukan samada fungsi yang telah dinyatakan dalam spesifikasi keperluan telah dilaksanakan di dalam sistem yang telah diintegrasikan tadi.



Ujian yang dijalankan juga melibatkan ujian baikpulih (*Recovery testing*) yang bertujuan menggagalkan sistem dan memastikan kegagalan tersebut dapat dipulihkan semula melalui masukan input yang tidak tepat oleh pengguna.

#### **6.2.4 Ujian Prestasi**

Ujian dijalankan secara masa larian (*run time*) bagi memastikan prestasi persembahan sistem secara keseluruhannya. Prestasi sistem diuji dari segi masa tindakbalas untuk S2PC memberikan contoh pengaturcaraan C, ketepatan jawapan dengan kehendak serta pilihan pengguna, kecekapan sistem dan ingatan yang digunakan.

#### **6.2.5 Ujian Penerimaan**

Apabila ujian prestasi selesai dilakukan, sistem akan disemak dan dirunding dengan pengguna samada ia memenuhi penerangan keperluan pelanggan serta berfungsi berdasarkan jangkaan pengguna.

#### **6.2.6 Ujian Pemasangan**

Sistem seterusnya akan diuji pemasangannya di persekitaran sebenar di mana ia akan digunakan. Pembangun akan menguji dan mencuba samada sistem boleh berjalan dengan lancar atau tidak dalam persekitaran

tersebut. Sekiranya sistem gagal untuk berfungsi seperti yang diinginkan, maka beberapa pengubahsuaian pada sistem atau persekitaran sistem perlu dilakukan agar sistem boleh berfungsi sebaik yang mungkin.

### 6.3 RUMUSAN

Melalui ujian-ujian yang telah dijalankan, risiko pada sistem dapat dikurangkan memandangkan ralat-ralat yang wujud dalam sistem ini dapat dikenalpasti dan kemudian diselesaikan dengan baik. Di samping itu pengujian program dan sistem ini memastikan kesemua fungsian sistem adalah memenuhi objektif dan keperluan Sistem Pemilihan Contoh Pengaturcaraan C



### 7.1 PENCAPAIAN OBJEKTIF

Fasa ini adalah bertujuan agar pembangun dapat menganalisa sejauh manakah kejayaan sistem yang dibangunkan dapat mencapai objektifnya di mana biasanya pembangun akan menerima maklumbalas daripada pengguna

Secara keseluruhannya Sistem Pemilihan Contoh Pengaturcaraan C berjaya mencapai objektif yang telah digariskan semasa fasa analisis dijalankan. S2PC telah mencapai matlamat utamanya iaitu sistem pintar yang menggunakan case-based reasoning sebagai satu kaedah dalam pembangunan sistemnya. Di samping itu S2PC juga berjaya dalam memudahkan lagi proses memilih contoh program pengaturcaraan C.

Namun sistem ini sebenarnya telah dilakukan beberapa perubahan daripada perancangan asalnya di dalam projek ilmiah tahap 1. Perubahan tersebut ialah:

➤ **Perubahan pada domain sistem dan beberapa matlamat**

Pada fasa analisis dijalankan sistem yang ingin dibangunkan ialah Sistem Pemilihan Contoh Pengaturcaraan Prolog (SPCP). Semasa fasa implementasi dijalankan pengaturcara mengalami kesukaran untuk mendapatkan contoh-contoh pengaturcaraan bagi Prolog sebagai case base. Setelah dipertimbangkan semula, pengaturcara telah mengambil

keputusan untuk mengubah matlamat projek di mana bahasa yang akan dijadikan domain bagi projek ini telah digantikan dengan bahasa C.

#### ➤ **Perisian yang digunakan**

Perisian pembangunan yang dicadangkan semasa projek ilmiah tahap 1 ialah WinProlog, tetapi pengaturcara mengalami kesukaran menguasai bahasa tersebut maka pengaturcara telah memilih Visual Basic 6.0 sebagai perisian pembangunan sistem.

#### ➤ **Ketiadaan beberapa fungsi**

Terdapat beberapa fungsi yang dirancang semasa fasa rekabentuk di dalam projek ilmiah tahap 1 tidak dapat diimplementasikan ke dalam sistem ini. Fungsi tersebut ialah fungsi bantuan dan fungsi laporan, di mana pengaturcara kekurangan masa untuk membangunkan kedua-dua fungsi tersebut. Sebagai ganti kepada fungsi bantuan pengaturcara telah memasukkan beberapa *tool tips* yang serba sedikit boleh membantu pengguna.

Terdapat juga beberapa fungsi yang telah dibuang iaitu fungsi menu yang membawa pengguna ke menu utama semula serta menu padam yang membolehkan pengguna memadamkan pencarian sebelumnya. Fungsi ini tidak lagi perlu diwujudkan kerana pengaturcara telah menggunakan



perisian Visual Basic yang menawarkan kemudahan yang lebih baik untuk membuat dan mengendalikan antaramuka.

Perubahan-perubahan yang telah dinyatakan di atas, walau bagaimanapun tetap tidak mengubah objektif utama projek ini di mana sistem ini tetap menggunakan case-based reasoning dan membantu pelajar dalam memilih contoh program pengaturcaraan. Ketidadaan beberapa fungsi juga tetap tidak menjejaskan keupayaan sistem dan projek ini masih mencapai objektifnya.

## 7.2 MASALAH PEMBANGUNAN SISTEM DAN PENYELESAIAN

Terdapat pelbagai masalah yang dihadapi oleh pengaturcara ataupun pembangun dalam menyiapkan sistem ini. Sesetengah masalah dapat diselesaikan dengan mudah manakala terdapat juga yang masalah yang tidak dapat diselesaikan yang memaksa kepada beberapa perubahan pada pembangunan sistem. Antara masalah-masalah yang wujud semasa pembangunan sistem ini ialah:

### ➤ Kekurangan bahan rujukan

Pembangun pada mulanya menghadapi masalah kekurangan bahan rujukan untuk membangunkan sistem ini. Rujukan yang diperlukan oleh

pembangun dalam sistem ini adalah berkaitan dengan case-based reasoning serta penggunaan perisian Visual Basic dan Microsoft Access. Kekurangan rujukan berkaitan dengan CBR adalah disebabkan ketiadaan buku rujukan yang banyak mengenainya di Malaysia, manakala perisian pula sumber rujukan yang terdapat di pasaran adalah sangat mahal dan buku rujukan yang terdapat di perpustakaan adalah sangat terhad dan tidak terkini.

Masalah ini dapat diatasi oleh pembangun dengan mengambil beberapa alternatif seperti menggunakan internet untuk mendapatkan maklumat mengenai CBR serta meminjam buku-buku berkaitan perisian daripada rakan-rakan. Perbincangan juga dilakukan dengan penasihat, beberapa pensyarah dan juga rakan-rakan bagi mendapatkan maklumat yang diperlukan.

#### ➤ **Kekurangan kemahiran dalam perisian**

Pembangun turut mengalami masalah kekurangan kemahiran di dalam bahasa pengaturcaraan Visual Basic dan Access. Kekurangan kemahiran ini mendatangkan masalah pada permulaan projek di mana pembangun perlu mempelajari bahasa tersebut terlebih dahulu dan ini agak memakan masa.



Pembangun mengatasi masalah ini dengan meminta tunjuk ajar daripada beberapa rakan yang mahir di samping banyak membuat rujukan dalam internet.

#### ➤ **Masalah melakukan *case scoring***

Kekurangan bahan rujukan menyebabkan pembangun mengalami masalah dalam melakukan *case-scoring* iaitu elemen penting dalam kaedah pencarian dalam CBR.

Masalah ini diatasi dengan berjumpa dengan pensyarah yang berpengetahuan mengenai CBR untuk mendapatkan pandangan dan pendapat.

### **7.3 PENILAIAN PENGGUNA**

Bagi memastikan bahawa S2PC ini berkesan, penilaian dengan pengguna telah dilakukan. Dalam penilaian ini sistem telah diuji oleh beberapa orang pelajar tahun satu daripada Fakulti Sains Komputer Universiti Malaya. Ini adalah kerana merekalah pengguna akhir (*end user*) kepada sistem ini di mana mereka sedang mengambil kursus WXES 1108 Asas Pengaturcaraan Komputer.

Penilai sistem ini perlu memberi penilaian terhadap beberapa aspek seperti rekabentuk antaramuka, mesra pengguna, kebolehpercayaan (dari segi ketepatan jawapan yang di beri) dan kebolehgunaan sistem.

Daripada respons penilai-penilai didapati mereka berpuas hati dengan S2PC. Antaramukanya yang mudah dan cara untuk memilih ciri-ciri pengaturcaraan yang dikehendaki juga mudah iaitu dengan hanya perlu menandakan ruangan *check box* yang dikehendaki. Pengguna juga merasakan sistem ini dapat membantu mereka dalam pembelajaran C di mana pemilihan contoh pengaturcaraan sebagai rujukan menjadi lebih mudah.

Walau bagaimanapun penilai-penilai tersebut merasakan bahawa sistem ini kekurangan contoh-contoh pengaturcaraan kerana beberapa pemilihan yang dilakukan oleh mereka menghasilkan keputusan yang kurang menepati kehendak mereka. Namun secara keseluruhannya sistem tersebut masih dipercayai oleh penilai tersebut untuk digunakan.

#### 7.4 KEKANGAN SISTEM

Sistem Pemilihan Contoh Pengaturcaraan C (S2PC) ini mempunyai beberapa kekangan yang telah dikenalpasti. Kekangan-kekangan yang dikenalpasti di dalam sistem ini ialah :



➤ **Kitar CBR tidak lengkap**

Kekangan utama sistem ini ialah kitar CBRnya yang tidak lengkap. Sebagaimana yang telah dinyatakan dalam bab 2 kajian literasi, pembangun telah mengambil keputusan untuk tidak melaksanakan adaptasi kes di dalam kitar CBR. Ini adalah kerana sistem ini memberikan contoh pengaturcaraan C sebagai *case solution*, maka adalah tidak sesuai dilakukan adaptasi terhadapnya kerana penyelesaian yang baru kemungkinan tidak sesuai dijadikan rujukan.

➤ **Pengguna sistem**

Pengguna sistem S2PC ini mestilah daripada mereka yang mempunyai pengetahuan-pengatahuan asas mengenai pengaturcaraan C disebabkan pengguna perlu memilih contoh berdasarkan ciri-ciri yang terdapat dalam bahasa C seperti pembolehubah, kawalan dan operator. Pengguna yang tidak mempunyai pengetahuan asas C mungkin tidak akan mempunyai idea untuk melakukan pemilihan contoh.

➤ **Bilangan kes**

Prestasi sebuah sistem CBR meningkat berdasarkan berapa banyak *case-base* yang dimilikinya dalam pangkalan pengetahuan. S2PC memiliki 100 kes di dalam pangkalan pengetahuannya maka jawapan

yang akan diperolehi oleh pengguna hanya akan berkisar kepada 100 kes ini sahaja.

#### ➤ Ciri-ciri pengaturcaraan untuk dipilih

S2PC menyenaraikan sebanyak 25 ciri sahaja dalam pengaturcaraan C untuk pengguna melakukan pemilihan. Beberapa ciri dalam pengaturcaraan C tidak dimasukkan seperti penunjuk dan struktur. Ini adalah kerana pembangun tidak mempunyai masa yang cukup untuk menyenaraikan kesemua ciri yang perlu dalam pembelajaran asas pengaturcaraan C.

## 7.5 CADANGAN MASA HADAPAN

#### ➤ Kitar CBR lengkap

Walaupun telah dinyatakan bahawa adaptasi kes tidak sesuai digunakan, namun pada masa hadapan mungkin dapat difikirkan kaedah melakukan adaptasi yang sesuai terhadap sistem ini supaya kitar CBRnya lengkap tanpa memberikan kesan terhadap jawapan yang akan diberikan kepada pengguna.



➤ **Penggunaan fungsi laporan**

Fungsi laporan yang digugurkan oleh pembangun boleh dibangunkan pada masa hadapan supaya pengguna memiliki kemudahan untuk membuat salinan kepada jawapan yang diperolehi.

➤ **Modul bantuan**

Fungsi *help* boleh diwujudkan untuk membantu pengguna menggunakan sistem ini tanpa bantuan daripada pihak lain khususnya pihak luar (pembangun) .

➤ **Pertambahan bilangan kes**

Sepertimana yang telah dinyatakan prestasi CBR akan bergantung kepada jumlah case-basanya, maka bilangan kes perlu ditambahkan pada masa akan datang. Penggunaan adaptasi kes juga dapat menambahkan bilangan kes di samping mewujudkan satu modul *update* yang membolehkan kes ditambahkan ke dalam pangkalan pengetahuan.

➤ **Pertambahan ciri-ciri untuk dipilih**

Ciri-ciri pengaturcaraan C boleh dikembangkan lagi sebagai contoh bagi ciri operator aritmetik pengguna boleh memilih lagi operator apa yang diingini sama ada +, -, /, \* ataupun %. Melalui cara ini jawapan yang diperolehi akan lebih menepati kehendak pengguna.

Pertambahan ciri-ciri bahasa C yang lain juga boleh dilakukan, sebagai contoh bagi ciri pembolehubah boleh ditambahkan ciri-ciri seperti unsigned int, unsigned long, continue sebagainya.

➤ **Sistem dihasilkan dalam bentuk web-based**

Bagi meluaskan lagi penggunaan S2PC pada masa akan datang, maka sistem ini dicadangkan supaya dimasukkan ke dalam web.

## 7.6 RUMUSAN

Berdasarkan segala masalah dan penyelesaian yang telah diambil oleh pembangun di samping penilaian-penilaian yang telah dilakukan, S2PC dapat dikatakan telah mencapai objektif utamanya iaitu penggunaan CBR sebagai kaedah penyelesaian masalah walaupun kitar CBRnya tidak lengkap.



## RUJUKAN

- [ 1 ] Aamodt, A, Plaza,E.(1994).Case-Based Reasoning:Foundational Issues,Methodological Variations and System Approaches. **AICom - Artificial Intelligence Communications**, Vol. 7: 1, 39-59.
- [ 2 ] Aha, D. , Kibler, D. , and Albert, M. K. (1991). Instance-Based Learning Algorithms. **Machine Learning**, vol.6 (1).
- [ 3 ] Brann,D.M,Thurman,D.A,Mitchell,C.M(1995).**Case-Based Reasoning as a Methodology for Accumulating Human Expertise for Discrete System Control**.Proceedings of the 1995 Int. Conf. on Systems,Man and Cybernatics.Vol 3:4219-4223.Vancouver,B.C.
- [ 4 ] Bratko,I (1990).**Prolog Programming for A.I**.Addison-Wesley
- [ 5 ] Brusilovsky, P., Specht, M., & Weber, G. (1995). Towards adaptive learning environments.**GISI 95** 322-329.
- [ 6 ] Burkhard,W.A.(2000).**C Untuk Pengaturcaraan**.Universiti Teknologi Malaysia.
- [ 7 ] Caellar,D.(2000),"**Teaching Programming:Some Lessons from Prolog**",Paper presented at LSTN-ICS 1st Annual Conference,Heriot-Watt University,Edinburgh.
- [ 8 ] Deitel,H.M,Deitel,P.J(2001).**C How to Program**,Third Edition,Prentice Hall.

- [ 9 ] Deitel,H.M,Deitel,P.J,Neito,T.R.(1999). **Visual Basic 6 How to Program**.Prentice Hall.
- [10 ] Choy,K.L, Lee,W.B(2000).Task allocation case-based reasoning for distributed manufacturing systems. **Logistics Information Management**,13 (3),167-176.
- [11] Kelley,A.,Pohl,I.(1998).**A Book on C**,Fourth Edition,Addison Wesley Longman.
- [12] Kolodner,J (1993). **Case-based reasoning**. Morgan Kaufmann.
- [13] Luger,G.F,Stubblefield,W.A.(1999).**Artificial Intelligence Structures and Strategies for Complex Problem Solving**,Third Edition: Addison Wesley Longman
- [14] Marini Abu Bakar,Norleyza Jailani,Sufian Idris(2000).**Pengaturcaraan C Edisi Revisi 2000**.Prentice Hall.
- [15] McIvor,R.T,Humphreys,P.K.(2000). A case-based reasoning approach to the make or buy decision.**Integrated Manufacturing Systems**,11 (5),295-310.
- [16] Mohamad Noorman Masrek,Kamarulariffin Abdul Jalil,Safawi Abdul Rahman(2001). **Analisis dan Rekabentuk Sistem Maklumat**,McGraw-Hill(Malaysia)
- [17] Pfleeger,S.L(2001).**Software Engineering Theory and Practice**,Second Edition,Prentice Hall.



- [18] Ribeiro, F.L. (2001). Project delivery system selection: a case-based reasoning framework. **Logistics Information Management**, 14 (5/6), 367-375.
- [19] Tan, H.H., D'Orazio, T.B. (1999). **C Programming for Engineering & Computer Science**. McGraw Hill.
- [20] Watson, I., Gardingen, D. (1999). **A Distributed Case-Based Reasoning Application for Engineering Sales Support**. In, Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), Vol 1: 600-605. Morgan Kaufmann Publishers Inc.
- [21] Weber, G. & Möllenberg, A. (1994). ELM-PE: A knowledge-based programming environment for learning LISP. **Proceedings of ED-MEDIA '94** 557-562.
- [22] Rujukan internet  
<http://www.ai-cbr.org>  
<http://www.cdi.uni-sb.de>  
<http://www.icsltsn.ac.uk>  
<http://www.swi-prolog.org>



## SENARAI LAMPIRAN

- LAMPIRAN 1:** MANUAL PENGGUNA S2PC
- LAMPIRAN 2:** JADUAL PERWAKILAN BAGI KES 1-30 DI DALAM PANGKALAN PENGETAHUAN
- LAMPIRAN 3:** JADUAL PENYELESAIAN KES (ATURCARA C) BAGI KES 1 – 8 DI DALAM PANGKALAN PENGETAHUAN
- LAMPIRAN 4:** KOD CASE SCORING
- LAMPIRAN 5:** BORANG SOAL-SELIDIK
- LAMPIRAN 6:** BORANG PENILAIAN PENGGUNA





## SISTEM PEMILIHAN CONTOH PENGALIRAN C (S2PC)

Versi 1.0.0

### 1.0. PEMERINTAH

Sistem Pemilihan Contoh Pengaliran C ataupun S2PC bukan merupakan program yang berdiri sendiri. Ia merupakan sebuah pengaliran (compiler) untuk pengaliran C.

# LAMPIRAN 1

## Manual Pengguna

### 2.0. KEPERLUAN HARDWARE S2PC

- > Processor: Intel Pentium atau AMD 200MHz atau lebih
- > Memori: 32 MB RAM
- > Carta kerdas: 4 GB
- > Tetikus
- > Papan Kekunci (Keyboard)
- > Monitor SVGA (cadangan resolusi 600x600)
- > Sistem Pengendalian Windows 93/95/2000



## **SISTEM PEMILIHAN CONTOH PENGATURCARAAN C ( S2PC )**

**Versi 1.0.0**

### **1.0 PEMBERITAHUAN**

Sistem Pemilihan Contoh Pengaturcaraan C ataupun S2PC bukan merupakan sistem pengajaran bahasa C. S2PC juga adalah bukan sebuah pengkompil (*compiler*) untuk pengaturcaraan C.

S2PC merupakan sebuah sistem yang berfungsi sebagai alat bantu pembelajaran bagi bahasa C. S2PC membantu dalam pemilihan contoh rujukan aturcara program C berdasarkan ciri-ciri yang dikehendaki oleh pengguna. Contoh-contoh aturcara program C yang terdapat dalam S2PC hanyalah merangkumi topik-topik asas dalam pembelajaran C.

### **2.0 KEPERLUAN MINIMUM S2PC**

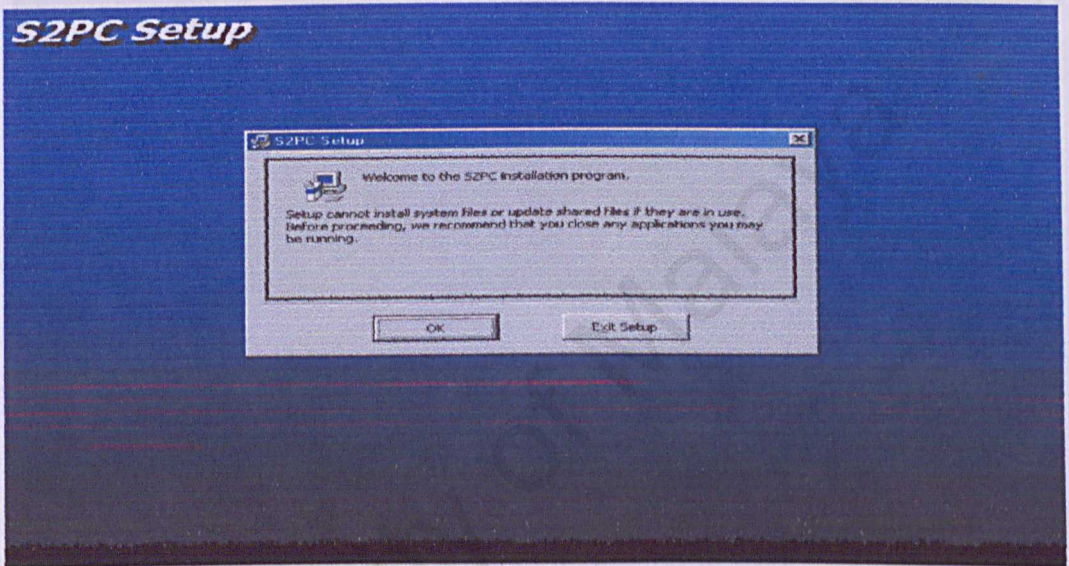
- Pemproses Intel Pentium atau AMD 200Mhz atau lebih
- Minima 32 MB RAM memori
- Cakera keras 4.0Gb
- Tetikus
- Papan Kekunci (*keyboard*)
- Monitor SVGA (cadangan resolusi 800\*600)
- Sistem Pengendalian Windows 98/Me/2000





### 3.0 PANDUAN PEMASANGAN SISTEM

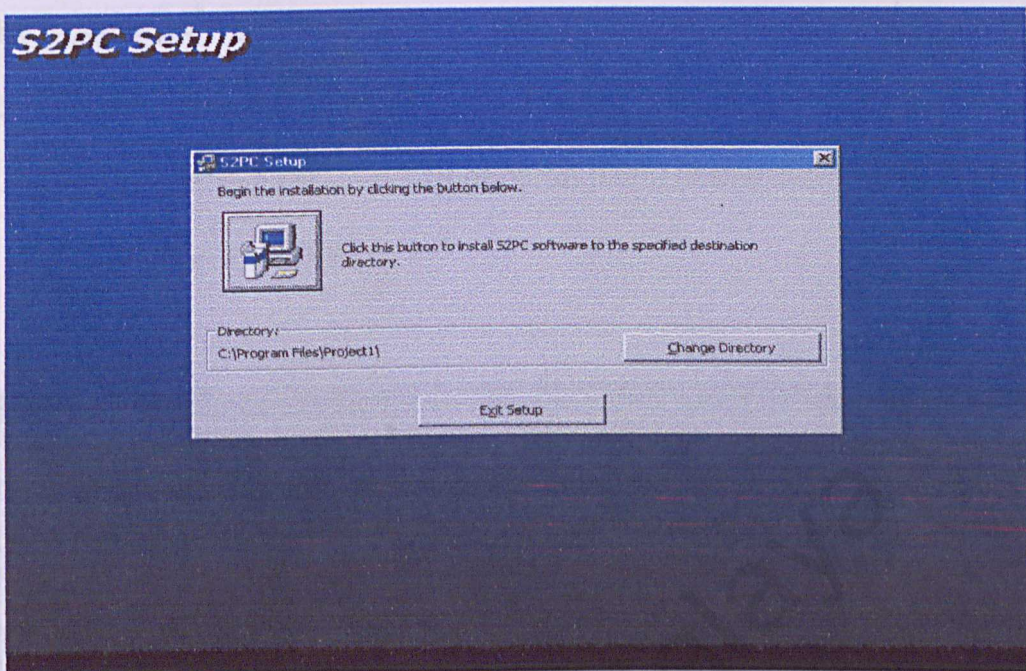
- Pengguna perlu memasukkan cakera padat ke dalam pemacu dan seterusnya klik pada ikon *setup.exe*. *Installation wizard (shield)* seperti di dalam rajah dibawah akan diaktifkan.



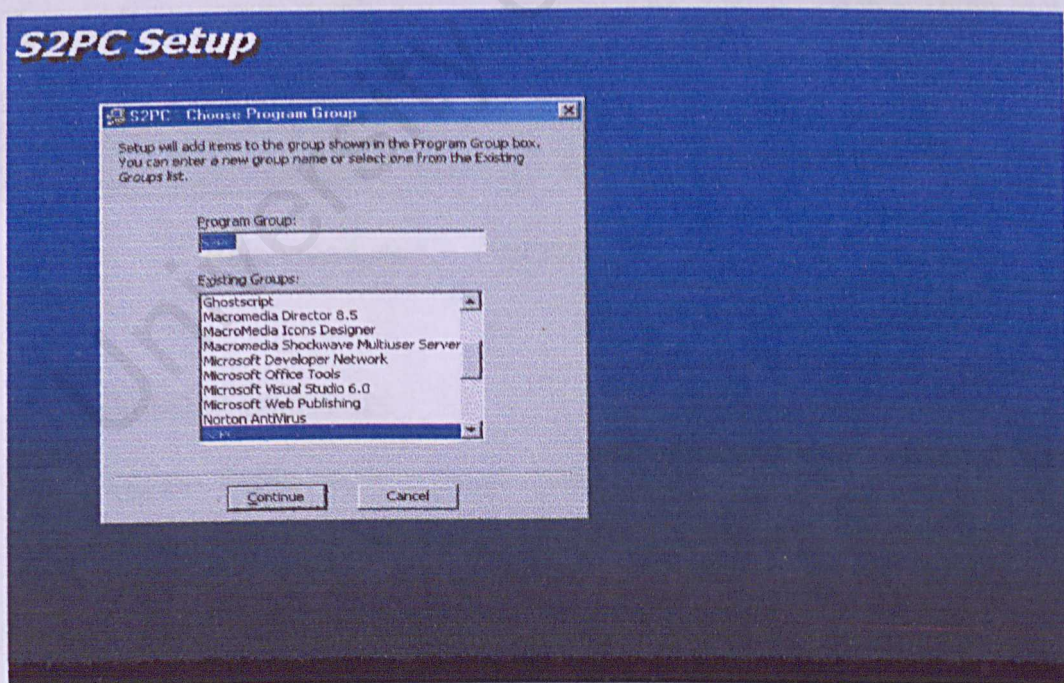
Rajah manual 1: Skrin *Setup*

- Klik pada butang OK dan seterusnya skrin pemilihan direktori akan dipaparkan [rajah manual 2]. Pilih direktori yang anda ingin sistem ini disimpan dan seterusnya klik pada ikon komputer.
- Berikutnya paparan *program groups* akan kelihatan [rajah manual 3]. Pilih *program groups* yang anda inginkan sistem ini berada.
- Tekan butang continue, seterusnya S2PC akan dimuatkan ke dalam komputer anda.





Rajah manual 2: Skrin pemilihan direktori



Rajah manual 3: Skrin penetapan *program groups*

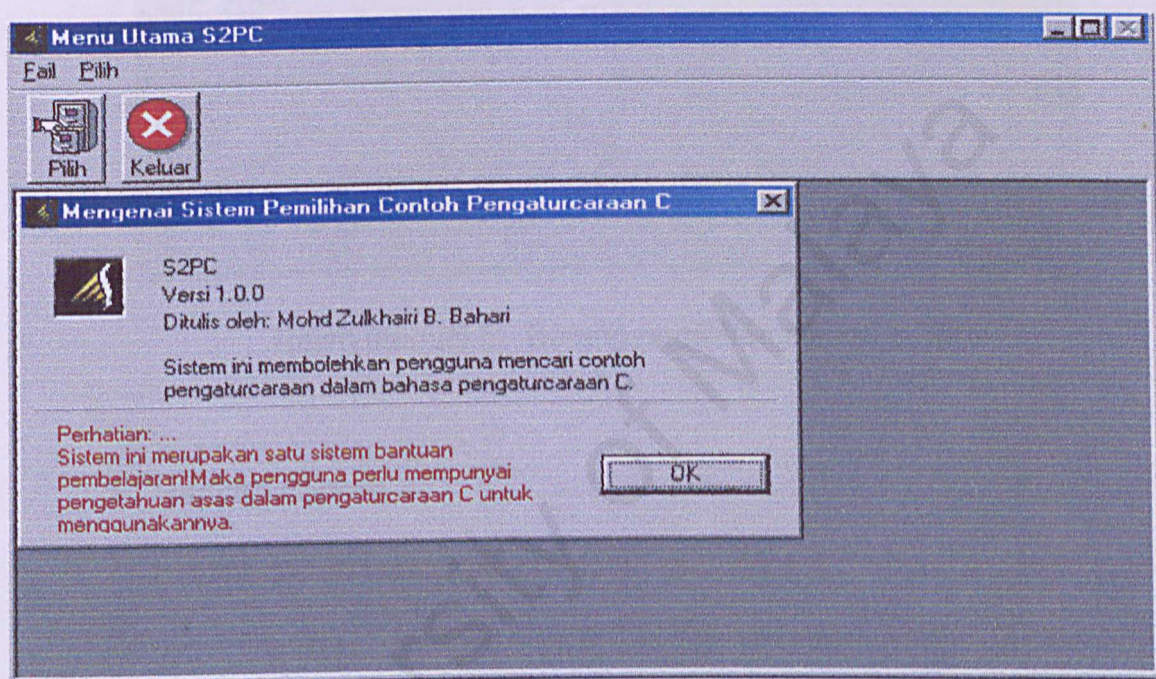




## 4.0 PENGGUNAAN SISTEM

### ➤ Melarikan sistem

Klik pada ikon S2PC.exe dan paparan menu utama S2PC seperti di bawah akan kelihatan.



Rajah manual 4: Menu utama S2PC

### ➤ Memulakan pemilihan

Pengguna boleh mula memilih contoh pengaturcaraan melalui klik pada ikon pilih ataupun menekan butang Alt + P pada papan kekunci

### ➤ Memilih ciri-ciri

Borang pemilihan akan dipaparkan dan pemilihan ciri yang dikehendaki boleh dilakukan dengan menanda pada *check box* yang berkaitan.





Rajah manual 5: Borang pemilihan contoh program

#### ➤ Memilih contoh

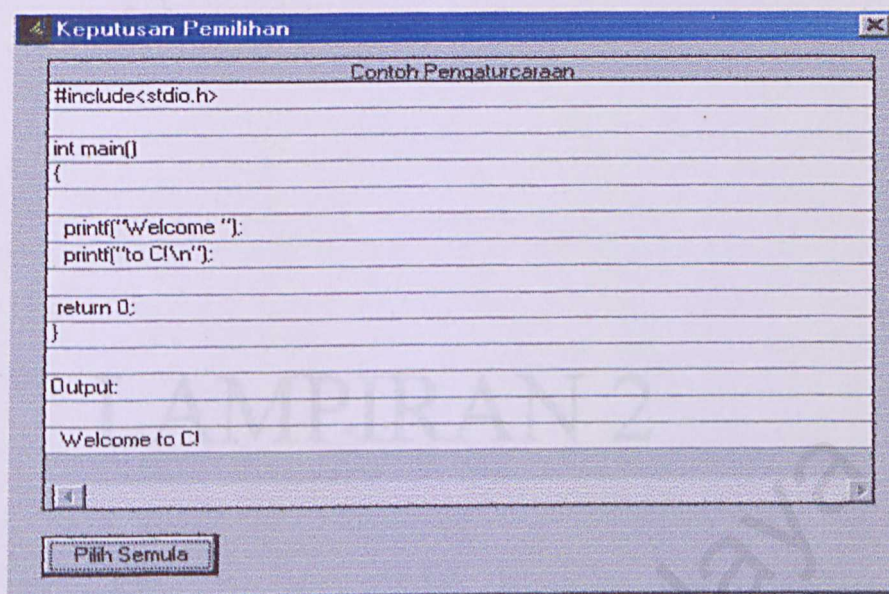
Setelah ciri-ciri yang dikehendaki ditanda tekan butang Mula Pilih pada borang pemilihan. Contoh pengaturcaraan yang dikehendaki akan dipaparkan dalam borang keputusan.

Pengguna boleh merujuk ciri-ciri yang telah dipilihnya untuk keputusan tersebut dengan mengaktifkan semula borang pemilihan

#### ➤ Pemilihan semula

Pengguna yang ingin memilih semula contoh pengaturcaraan boleh melakukannya dengan menekan butang pilih semula pada borang keputusan.

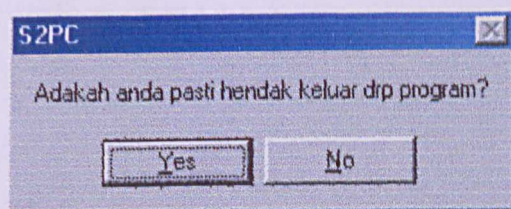




Rajah manual 6: Paparan keputusan bagi pemilihan

#### ➤ Keluar daripada sistem

Pengguna boleh keluar dari sistem pada bila-bila masa dengan melakukan klik pada ikon keluar ataupun menekan Alt + F + K. Kotak mesej akan dipaparkan dan pengguna akan keluar sekiranya butang Yes ditekan.



Rajah manual 7: Mesej keluar daripada sistem

## LAMPIRAN 2

Jadual Perwakilan Bagi Kes Di Dalam  
Pangkalan Pengetahuan



case	printf/scanf	getchar/putchar	gets/puts	int	float	long	char	double
1	1	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0
3	1	0	0	1	0	0	0	0
4	1	0	0	1	0	0	0	0
5	1	0	0	1	1	0	0	0
6	1	0	0	1	0	0	0	0
7	1	0	0	1	0	0	0	0
8	1	0	0	1	0	0	0	0
9	1	0	0	1	0	0	0	0
10	1	0	0	1	0	0	0	0
11	1	0	0	1	0	0	0	1
12	1	0	0	1	0	0	0	0
13	1	0	0	1	0	0	0	0
14	1	0	0	1	0	0	0	0
15	1	0	0	1	0	0	1	0
16	1	0	0	1	0	0	0	0
17	1	0	0	1	0	0	0	0
18	1	0	0	1	0	0	0	0
19	1	0	0	1	0	1	0	0
20	1	0	0	1	0	1	0	0
21	1	0	0	1	0	0	0	0
22	1	0	0	1	0	0	0	0
23	1	0	0	1	0	0	0	0
24	1	0	0	1	0	0	0	0
25	1	0	0	1	0	0	1	0
26	1	0	0	1	0	0	0	0
27	1	0	0	0	0	0	0	0
28	1	0	0	1	0	0	0	0
29	1	0	0	1	0	0	0	0
30	1	0	0	1	0	0	0	0





[illegible]

## LAMPIRAN 3

Jadual Penyelesaian Bagi Kes Di Dalam  
Pangkalan Pengetahuan



ID1	id	kod
1	ID	kod
2	1	#include<stdio.h>
3	1	
4	1	int main()
5	1	{
6	1	
7	1	printf("Welcome ");
8	1	printf("to C!\n");
9	1	
10	1	return 0;
11	1	}
12	1	
13	1	Output:
14	1	
15	1	Welcome to C!
16	2	#include<stdio.h>
17	2	
18	2	int main()
19	2	{
20	2	
21	2	int integer1, integer2, sum;
22	2	
23	2	printf("Enter first integer\n");
24	2	scanf("%d",&integer1);
25	2	printf("Enter second integer\n");
26	2	scanf("%d",&integer2);
27	2	sum=integer1+integer2;
28	2	printf("sum is %d\n",sum);
29	2	
30	2	return 0;
31	2	}
32	2	
33	2	Output:
34	2	
35	2	Enter first integer
36	2	45
37	2	Enter second integer
38	2	72
39	2	sum is 117
40	3	#include<stdio.h>
41	3	int main()
42	3	{
43	3	
44	3	int num1,num2;
45	3	
46	3	printf("Enter two integers, and I will tell you\n");

ID1	id	kod
47 3		printf("the relationships they satisfy: ");
48 3		scanf("%d%d",&num1,&num2);
49 3		
50 3		if(num1==num2)
51 3		printf("%d is equal to %d\n",num1,num2);
52 3		
53 3		if(num1!=num2)
54 3		printf("%d is not equal to %d\n",num1,num2);
55 3		
56 3		if(num1<num2)
57 3		printf("%d is less than %d\n",num1,num2);
58 3		
59 3		if(num1>num2)
60 3		printf("%d is greater than %d\n",num1,num2);
61 3		
62 3		if(num1<=num2)
63 3		printf("%d is less than or equal to %d\n",num1,num2);
64 3		
65 3		if(num1>=num2)
66 3		printf("%d is greater than or equal to %d\n",num1,num2);
67 3		
68 3		return 0;
69 3		}
70 3		
71 3		
72 3		Output:
73 3		
74 3		1) Enter two integers,and I will tell you
75 3		the relationships they satisfy: 3 7
76 3		3 is not equal to 7
77 3		3 is less than 7
78 3		3 is less than or equal to 7
79 3		
80 3		2) Enter two integers,and I will tell you
81 3		the relationships they satisfy: 22 12
82 3		22 is not equal to 12
83 3		22 is greater than 12
84 3		22 is greater than or equal to 12
85 4		#include<stdio.h>
86 4		
87 4		int main()
88 4		{
89 4		
90 4		int counter,grade,total,average;
91 4		
92 4		total=0;



ID1	id	kod
93 4		counter=1;
94 4		
95 4		while(counter<=10){
96 4		printf("Enter grade: ");
97 4		scanf("%d",&total+grade);
98 4		total=total+grade;
99 4		counter=counter+1;
100 4		}
101 4		
102 4		average=total/10;
103 4		printf("Class average is %d\n,average);
104 4		
105 4		return 0;
106 4		}
107 4		
108 4		Output:
109 4		
110 4		Enter grade: 98
111 4		Enter grade: 76
112 4		Enter grade: 71
113 4		Enter grade: 87
114 4		Enter grade: 83
115 4		Enter grade: 90
116 4		Enter grade: 57
117 4		Enter grade: 79
118 4		Enter grade: 82
119 4		Enter grade: 94
120 4		Class average is 81;
121 5		#include <stdio.h>
122 5		
123 5		int main()
124 5		{
125 5		float average;
126 5		int counter,grade,total;
127 5		
128 5		total=0;
129 5		counter=0;
130 5		
131 5		printf("Enter grade,-1 to end:");
132 5		scanf("%d",&grade);
133 5		
134 5		while (grade !=-1){
135 5		total = total + grade;
136 5		counter = counter + 1;
137 5		printf("Enter grade, -1 to end:");
138 5		scanf("%d",&grade);

ID1	id	kod
139 5	}	
140 5		
141 5	if (counter !=0){	
142 5	average = (float) total/counter;	
143 5	printf("Class average is %.2f",average);	
144 5	}	
145 5	else	
146 5	printf("No grades were entered\n");	
147 5	return 0;	
148 5	}	
149 5		
150 5	output:	
151 5	Enter grade, -1 to end: 75	
152 5	Enter grade, -1 to end: 94	
153 5	Enter grade, -1 to end: 97	
154 5	Enter grade, -1 to end: 88	
155 5	Enter grade, -1 to end: 70	
156 5	Enter grade, -1 to end: 64	
157 5	Enter grade, -1 to end: 83	
158 5	Enter grade, -1 to end: 89	
159 5	Enter grade, -1 to end: -1	
160 5	Class average is 82.50	
161 6	#include<stdio.h>	
162 6	int main()	
163 6	{	
164 6	int passes=0,failures=0,student=1,result;	
165 6		
166 6	while(student<=10){	
167 6	printf("Enter result(1=pass,2=fail):");	
168 6	scanf("%d",&result);	
169 6		
170 6	if(result==1)	
171 6	passes=passes+1;	
172 6	else	
173 6	failures=failures+1;	
174 6	student=student+1;	
175 6	}	
176 6	printf("Passed %d\n",passes);	
177 6	printf("Failed %d\n",failures);	
178 6		
179 6	if(passes>8)	
180 6	printf("Raise tuition\n");	
181 6	return 0;	
182 6	}	
183 6		
184 6	output:	



ID1	id	kod
185 6		Enter grade, -1 to end:1
186 6		Enter grade, -1 to end:1
187 6		Enter grade, -1 to end:1
188 6		Enter grade, -1 to end:2
189 6		Enter grade, -1 to end:1
190 6		Enter grade, -1 to end:1
191 6		Enter grade, -1 to end:1
192 6		Enter grade, -1 to end:1
193 6		Enter grade, -1 to end:1
194 6		Enter grade, -1 to end:1
195 6		Passed 9
196 6		Failed 1
197 6		Raise tuition
198 7		#include<stdio.h>
199 7		int main()
200 7		{
201 7		int c=5;
202 7		
203 7		printf("%d\n",c);
204 7		printf("%d\n",c++);
205 7		printf("%d\n\n",c);
206 7		
207 7		c=5;
208 7		
209 7		printf("%d\n",c);
210 7		printf("%d\n",++c);
211 7		printf("%d\n",c);
212 7		return 0;
213 7		}
214 7		
215 7		output:
216 7		5
217 7		5
218 7		6
219 7		
220 7		5
221 7		6
222 7		6
223 8		#include<stdio.h>
224 8		
225 8		int main()
226 8		{
227 8		int counter=1;
228 8		
229 8		while(counter<=10){
230 8		printf("%d\n",counter);

## KOD CASE SCORING

Private Sub btnAggihan\_Click()

Dim pangkalanCRK Bernilai

Dim score As Integer

Dim temp\_score As Integer

Dim jawaban As Integer

score = 0

Model Referensi: rujukan kepada pangkalan data

With ActiveSheet

MoveFirst

Do While Not IsEmpty

temp\_score = 0

For scoring

Select Case Fields("print/scan") Value

Case 1

If print Value = 1 Then

temp\_score

Else

temp\_score = temp\_score + 0

End If

Case 0

If print Value = 0 Then

temp\_score = temp\_score + 1

Else

temp\_score = temp\_score + 0

End If

End Select

Select Case Fields("getchar/scan") Value

Case 1

If getchar Value = 1 Then

temp\_score = temp\_score + 3

Else

temp\_score = temp\_score + 0

End If

Case 0

If getchar Value = 0 Then

temp\_score = temp\_score + 1

Else

temp\_score = temp\_score + 0

End If

End Select

Select Case Fields("get/pair") Value

## LAMPIRAN 4

### Kod Case Scoring



## KOD CASE SCORING

Private Sub mulapilih\_Click()  
'enjin pencarian CBR bermula

Dim score As Integer

Dim temp score As Integer

Dim jawapan As Integer

score = 0

Adodc1.Refresh 'rujukan kepada pangkalan data

With Adodc1.Recordset

.MoveFirst

Do While Not .EOF

temp score = 0

'case scoring

Select Case .Fields("printf/scanf").Value

Case 1

If printf.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If printf.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("getchar/putchar").Value

Case 1

If getchar.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If getchar.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("gets/puts").Value

Case 1

If gets.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If gets.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("int").Value

Case 1

If ints.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If ints.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("float").Value

Case 1

If float.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If float.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("long").Value



Case 1

If loong.Value = 1 Then  
    tempscore = tempscore + 3  
Else  
    tempscore = tempscore + 0  
End If

Case 0

If loong.Value = 0 Then  
    tempscore = tempscore + 1  
Else  
    tempscore = tempscore + 0  
End If

End Select

Select Case .Fields("char").Value

Case 1

If char.Value = 1 Then  
    tempscore = tempscore + 3  
Else  
    tempscore = tempscore + 0  
End If

Case 0

If char.Value = 0 Then  
    tempscore = tempscore + 1  
Else  
    tempscore = tempscore + 0  
End If

End Select

Select Case .Fields("double").Value

Case 1

If doubles.Value = 1 Then  
    tempscore = tempscore + 3  
Else  
    tempscore = tempscore + 0  
End If

Case 0

If doubles.Value = 0 Then  
    tempscore = tempscore + 1  
Else  
    tempscore = tempscore + 0  
End If

End Select

Select Case .Fields("hubungan").Value

Case 1

```
If hubungan.Value = 1 Then
    tempcore = tempcore + 3
Else
    tempcore = tempcore + 0
End If
```

Case 0

```
If hubungan.Value = 0 Then
    tempcore = tempcore + 1
Else
    tempcore = tempcore + 0
End If
```

End Select

Select Case .Fields("mantik").Value

Case 1

```
If mantik.Value = 1 Then
    tempcore = tempcore + 3
Else
    tempcore = tempcore + 0
End If
```

Case 0

```
If mantik.Value = 0 Then
    tempcore = tempcore + 1
Else
    tempcore = tempcore + 0
End If
```

End Select

Select Case .Fields("aritmetik").Value

Case 1

```
If aritmetik.Value = 1 Then
    tempcore = tempcore + 3
Else
    tempcore = tempcore + 0
End If
```

Case 0

```
If aritmetik.Value = 0 Then
    tempcore = tempcore + 1
Else
    tempcore = tempcore + 0
End If
```

End Select

Select Case .Fields("tokok dan susut").Value

Case 1

```
If tokok.Value = 1 Then
```



```
    tempcore = tempcore + 3
Else
    tempcore = tempcore + 0
End If
Case 0
    If tokok.Value = 0 Then
        tempcore = tempcore + 1
    Else
        tempcore = tempcore + 0
    End If
End Select
Select Case .Fields("if/else").Value
```

```
Case 1
    If ifelse.Value = 1 Then
        tempcore = tempcore + 3
    Else
        tempcore = tempcore + 0
    End If
Case 0
    If ifelse.Value = 0 Then
        tempcore = tempcore + 1
    Else
        tempcore = tempcore + 0
    End If
End Select
Select Case .Fields("switch").Value
```

```
Case 1
    If switc.Value = 1 Then
        tempcore = tempcore + 3
    Else
        tempcore = tempcore + 0
    End If
Case 0
    If switc.Value = 0 Then
        tempcore = tempcore + 1
    Else
        tempcore = tempcore + 0
    End If
End Select
Select Case .Fields("break").Value
```

```
Case 1
    If brek.Value = 1 Then
        tempcore = tempcore + 3
```

```
Else  
    tempscore = tempscore + 0  
End If  
Case 0  
    If brek.Value = 0 Then  
        tempscore = tempscore + 1  
    Else  
        tempscore = tempscore + 0  
    End If  
End Select  
Select Case .Fields("for").Value
```

```
Case 1  
    If fore.Value = 1 Then  
        tempscore = tempscore + 3  
    Else  
        tempscore = tempscore + 0  
    End If  
Case 0  
    If fore.Value = 0 Then  
        tempscore = tempscore + 1  
    Else  
        tempscore = tempscore + 0  
    End If  
End Select  
Select Case .Fields("while").Value
```

```
Case 1  
    If whil.Value = 1 Then  
        tempscore = tempscore + 3  
    Else  
        tempscore = tempscore + 0  
    End If  
Case 0  
    If whil.Value = 0 Then  
        tempscore = tempscore + 1  
    Else  
        tempscore = tempscore + 0  
    End If  
End Select  
Select Case .Fields("do").Value
```

```
Case 1  
    If doo.Value = 1 Then  
        tempscore = tempscore + 3  
    Else
```



```
    tempscore = tempscore + 0
End If
Case 0
    If doo.Value = 0 Then
        tempscore = tempscore + 1
    Else
        tempscore = tempscore + 0
    End If
End Select
Select Case .Fields("tatasusunan 1d").Value
```

```
Case 1
    If tsn1d.Value = 1 Then
        tempscore = tempscore + 3
    Else
        tempscore = tempscore + 0
    End If
```

```
Case 0
    If tsn1d.Value = 0 Then
        tempscore = tempscore + 1
    Else
        tempscore = tempscore + 0
    End If
End Select
Select Case .Fields("tatasusunan 2d").Value
```

```
Case 1
    If tsn2d.Value = 1 Then
        tempscore = tempscore + 3
    Else
        tempscore = tempscore + 0
    End If
```

```
Case 0
    If tsn2d.Value = 0 Then
        tempscore = tempscore + 1
    Else
        tempscore = tempscore + 0
    End If
End Select
Select Case .Fields("rentetan").Value
```

```
Case 1
    If rentetan.Value = 1 Then
        tempscore = tempscore + 3
    Else
        tempscore = tempscore + 0
```

```
End If
Case 0
  If rentetan.Value = 0 Then
    tempscore = tempscore + 1
  Else
    tempscore = tempscore + 0
  End If
End Select
Select Case .Fields("pengumpulan unsur").Value
```

```
Case 1
  If umpuk.Value = 1 Then
    tempscore = tempscore + 3
  Else
    tempscore = tempscore + 0
  End If
```

```
Case 0
  If umpuk.Value = 0 Then
    tempscore = tempscore + 1
  Else
    tempscore = tempscore + 0
  End If
End Select
Select Case .Fields("fungsi").Value
```

```
Case 1
  If fungsi.Value = 1 Then
    tempscore = tempscore + 3
  Else
    tempscore = tempscore + 0
  End If
```

```
Case 0
  If fungsi.Value = 0 Then
    tempscore = tempscore + 1
  Else
    tempscore = tempscore + 0
  End If
```

```
End Select
Select Case .Fields("maths").Value
```

```
Case 1
  If mathh.Value = 1 Then
    tempscore = tempscore + 3
  Else
    tempscore = tempscore + 0
  End If
```



Case 0

If mathh.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

Select Case .Fields("define").Value

Case 1

If dfine.Value = 1 Then

temp score = temp score + 3

Else

temp score = temp score + 0

End If

Case 0

If dfine.Value = 0 Then

temp score = temp score + 1

Else

temp score = temp score + 0

End If

End Select

If temp score > score Then

score = temp score

jawapan = .Fields("case").Value

txtjawapan.Text = jawapan

End If

.MoveNext

Loop

End With

frmkeputusan.Show

mulapilih.Enabled = False

hilangcheck

End Sub

# LAMPIRAN 5

## Borang Soal Selidik

- |                                   |                                 |                                 |
|-----------------------------------|---------------------------------|---------------------------------|
| <input type="checkbox"/> Maderela | <input type="checkbox"/> Daring | <input type="checkbox"/> Papan  |
| <input type="checkbox"/> Puz      | <input type="checkbox"/> Kertas | <input type="checkbox"/> Kertas |
| <input type="checkbox"/> Gelas    | <input type="checkbox"/> Kertas | <input type="checkbox"/> Kertas |

1. Dalam konteks kajian, anda akan lakukan kajian yang akan dijalankan di atas, untuk kajian yang akan dijalankan.
- ☐ Menjalankan kajian yang akan dijalankan di atas di dalam kajian yang akan dijalankan.
- ☐ Menjalankan kajian yang akan dijalankan di atas di dalam kajian yang akan dijalankan.
- ☐ Menjalankan kajian yang akan dijalankan di atas di dalam kajian yang akan dijalankan.
- ☐ Lain-lain (jika ada) \_\_\_\_\_

2. Dalam konteks kajian, anda akan lakukan kajian yang akan dijalankan di atas, untuk kajian yang akan dijalankan.

3. Dalam konteks kajian, anda akan lakukan kajian yang akan dijalankan di atas, untuk kajian yang akan dijalankan.

4. Dalam konteks kajian, anda akan lakukan kajian yang akan dijalankan di atas, untuk kajian yang akan dijalankan.



Borang ini mempunyai dua bahagian iaitu :  
 Bahagian A untuk diisi oleh semua pelajar  
 Bahagian B untuk diisi oleh pelajar FSKTM sahaja  
 Bahagian C bahagian tambahan untuk pelajar dari FSKTM Jabatan  
 Kepintaran Buatan

## BAHAGIAN A ( diisi oleh semua )

Sila baca arahan dengan teliti

JANTINA : \_\_\_\_\_ TAHUN PENGAJIAN : \_\_\_\_\_  
 FAKULTI : \_\_\_\_\_  
 JABATAN : \_\_\_\_\_

Bagi soalan di bawah potong yang mana tidak berkenaan

1. Pernahkah anda menggunakan contoh penyelesaian di dalam buku sebagai panduan untuk menjawab soalan-soalan dalam tugas ataupun tutorial.  
 (contohnya anda merujuk contoh penyelesaian masalah Matematik dalam buku untuk menjawab soalan-soalan dalam tugas Matematik anda).

Ya / Tidak

Bagi soalan berikut sila tandakan ✓ pada petak yang berkenaan. Anda boleh menanda lebih daripada satu

2. Jika anda pernah menggunakan kaedah merujuk contoh dalam menyelesaikan masalah yang diberi, dalam bidang apakah anda sering kali menggunakan kaedah ini.

<input type="checkbox"/> Matematik	<input type="checkbox"/> Biologi	<input type="checkbox"/> Perakaunan
<input type="checkbox"/> Fizik	<input type="checkbox"/> Kejuruteraan	<input type="checkbox"/> Lain-lain (nyatakan)
<input type="checkbox"/> Kimia	<input type="checkbox"/> Sains Komputer	_____

Bagi soalan 3 sila susunkan maklumat mengikut keutamaan. Berikan nombor 1 bagi yang mempunyai keutamaan yang tinggi, 2 bagi keutamaan kedua dan seterusnya di dalam petak yang disediakan

3. Dalam membuat tugas, tutorial ataupun latihan makmal bagi kursus yang anda nyatakan di atas, apakah kaedah yang anda gunakan mengikut keutamaan.

<input type="checkbox"/>	Menggunakan contoh-contoh penyelesaian sedia ada di dalam buku atau bahan rujukan lain sebagai panduan.
<input type="checkbox"/>	Membuat tugas tersebut secara "try and error".
<input type="checkbox"/>	Meminta nasihat dan membuat rujukan dengan pensyarah atau tutor.
<input type="checkbox"/>	Lain-lain ( nyatakan ) _____

4. Sejauh manakah pendapat anda, sekiranya dikatakan bahawa kaedah merujuk contoh dalam rujukan adalah kaedah pembelajaran yang selalu digunakan pelajar dalam menyelesaikan serta memahami permasalahan (tugas/tutorial) yang diberi

Setuju / Tidak Setuju

Nyatakan sebab-sebab bagi jawapan anda.

---



---

Sekiranya anda bukan pelajar FSKTM soal-selidik tamat disini.



Bagi soalan berikut sila tandakan  $\checkmark$  pada petak yang berkenaan

5. Sila nyatakan bahasa pengaturcaraan apakah yang pernah anda pelajari di dalam program pengajian anda di FSKTM.

☐

C

☐

C ++

☐

Java

☐

Visual Basic

☐

Prolog

☐

Lain-lain ( nyatakan )

6. Berdasarkan bahasa-bahasa pengaturcaraan yang telah anda pelajari, nyatakan bahasa pengaturcaraan yang paling anda gemari dan jelaskan sebab-sebabnya.

---

---

7. Berdasarkan bahasa-bahasa pengaturcaraan yang telah anda pelajari, nyatakan bahasa yang pengaturcaraan yang paling sukar dipelajari dan jelaskan sebab-sebabnya.

---

---

Bagi soalan 8 sila susunkan maklumat mengikut keutamaan. Berikan nombor 1 bagi yang mempunyai keutamaan yang tinggi, 2 bagi keutamaan kedua dan seterusnya di dalam petak yang disediakan

8. Dalam memahami serta menguasai sesuatu bahasa pengaturcaraan, apakah kaedah yang anda gunakan mengikut keutamaan.

☐

Menggunakan contoh-contoh program sedia ada di dalam buku atau bahan rujukan lain sebagai panduan.

☐

Membuat program tersebut secara " try and error " .

☐

Meminta nasihat dan membuat rujukan dengan pensyarah atau tutor.

☐

Lain-lain ( nyatakan ) \_\_\_\_\_

Bagi soalan di bawah potong yang mana tidak berkenaan

9. Sekiranya terdapat sebuah sistem yang boleh membantu pelajar dalam memilih contoh program yang sesuai untuk dijadikan rujukan dalam membuat tugas pengaturcaraan, adakah anda berminat untuk menggunakannya.

Ya / Tidak

Nyatakan sebab-sebab bagi jawapan anda.

---

---

10. Jika terdapat sebuah sistem yang ingin dibangunkan, bertujuan meningkatkan kefahaman pelajar dalam sesuatu bahasa pengaturcaraan, pada pendapat anda sistem yang bagaimanakah harus dibangunkan.

---

---

Sekiranya anda bukan pelajar dari Jabatan Kepintaran Buatan, soal-selidik tamat di sini.



Bagi soalan 12 dan 13 potong yang mana tidak berkenaan

11. Sudahkah anda mengambil kursus Pengaturcaraan Logik.

Ya / Tidak

Sekiranya jawapan anda tidak, soal-selidik tamat di sini.

12. Pada pendapat anda adakah Prolog mudah dipelajari

Ya / Tidak

Bagi soalan berikut sila tandakan  $\checkmark$  pada petak yang berkenaan. Anda boleh menanda lebih daripada satu

13. Sekiranya anda berpendapat Prolog sukar dipelajari, nyatakan sebab-sebabnya

☐

Sintaks dan gaya pengaturcaraan yang berbeza dengan bahasa pengaturcaraan yang lain

☐

Kekurangan bahan-bahan rujukan

☐

Kesukaran untuk mendapatkan contoh-contoh pengaturcaraan yang sesuai sebagai panduan

☐

Kesukaran untuk memilih contoh program yang sepadan dengan masalah ( tugas ) yang diberi, untuk dijadikan panduan.

☐

Lain-lain ( nyatakan ) \_\_\_\_\_

14. Sila nyatakan apakah bahagian-bahagian yang sukar anda pelajari di dalam kursus Pengaturcaraan Logik.

---

---

Soal-selidik tamat.

NAMA \_\_\_\_\_

JUJUTERA \_\_\_\_\_ TAHUN PENGAJIAN \_\_\_\_\_

JABATAN \_\_\_\_\_

## LAMPIRAN 6

### Borang Penilaian Pengguna



**BORANG PENILAIAN PENGGUNA SISTEM S2PC**

**MAKLUMAT RESPONDAN**

NAMA : \_\_\_\_\_

JANTINA : \_\_\_\_\_ TAHUN PENGAJIAN : \_\_\_\_\_

JABATAN : \_\_\_\_\_

**SOALAN-SOALAN**

1. Bagi setiap ciri di bawah, berikan nilai di antara 1-5 mengikut skala di bawah

1	2	3	4	5
sangat tidak memuaskan	tidak memuaskan	memuaskan	baik	sangat baik
Rekabentuk antaramuka				<input type="checkbox"/>
Mesra pengguna				<input type="checkbox"/>
Kebolehpercayaan (dari segi ketepatan jawapan yang di beri)				<input type="checkbox"/>
Kebolehgunaan sistem.				<input type="checkbox"/>

2. Pada pandangan anda adakah sistem ini sesuai digunakan dalam memilih contoh pengaturcaraan C yang sesuai untuk dijadikan rujukan.

\_\_\_\_\_

\_\_\_\_\_

3. Pada pandangan anda adakah sistem ini dapat memberi contoh rujukan yang dikehendaki oleh anda. (sebagaimana yang anda minta dalam pencarian).

\_\_\_\_\_

\_\_\_\_\_

4. Sila nyatakan komen anda terhadap keseluruhan sistem ini.

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_